CS402 Data Mining & Warehousing

MODULE 6

April 30, 2020

Outline:

Hierarchical Clustering methods I

- A hierarchical clustering method works by grouping data objects into a tree of clusters
- Furthur classified as: **agglomerative** *or* **divisive**
- It produces clusters in which the clusters at each level of the hierarchy are created by merging clusters at the next lower level.
- At the lowest level, each cluster contains a single observation.
- At the highest level there is only one cluster containing all of the data
- **Agglomerative method (bottom-up method)**
  - This bottom-up strategy starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all of the objects are in a single cluster or until certain termination conditions are satisfied
  - Start at the bottom and at each level recursively merge a selected pair of clusters into a single cluster
  - This produces a grouping at the next higher level with one less cluster.

Hierarchical Clustering methods II

- If there are N observations in the dataset, there will be $N-1$ levels in the hierarchy.
- The pair chosen for **merging** consist of the two groups with the **"smallest intercluster dissimilarity"**.
- ie; They differ only in their definition of intercluster similarity.

- **Divisive method (top-down method)**
  - Does the reverse of agglomerative hierarchical clustering by starting with all objects in one cluster.
  - It subdivides the cluster into smaller and smaller pieces, until each object forms a cluster on its own or until it satisfies certain termination conditions
  - The termination condition could be a desired number of clusters is obtained or the diameter of each cluster is within a certain threshold
  - It starts at the top and at each level recursively split one of the existing clusters at that level into two new clusters.
  - If there are N observations in the dataset, there the divisive method also will produce $N-1$ levels in the hierarchy.

Hierarchical Clustering methods III

- The **split** is chosen to produce two new groups with the **"largest inter-cluster dissimilarity"**.

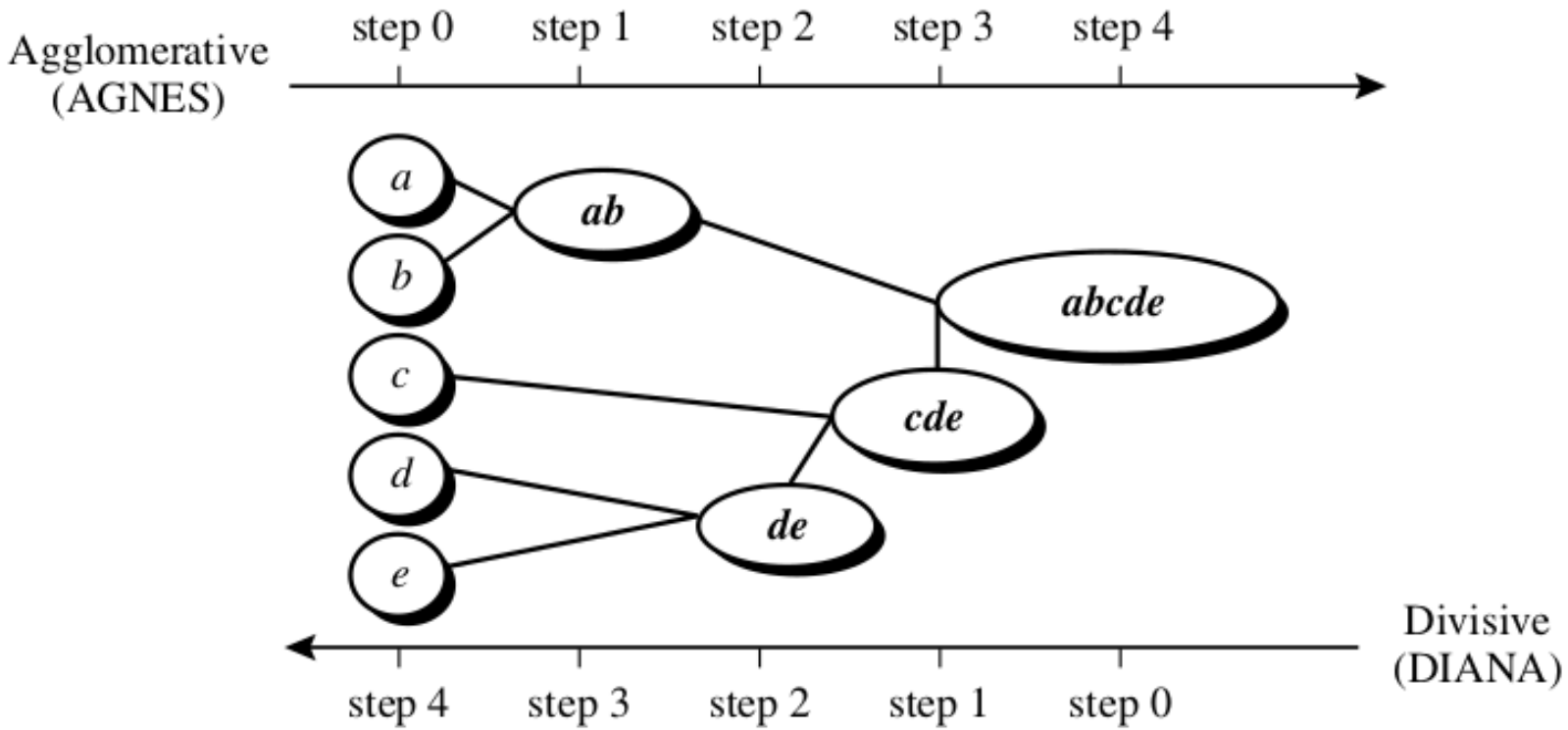


Figure: Agglomerative and divisive hierarchical clustering on data objects $\{a, b, c, d, e\}$

Hierarchical Clustering methods IV

Measures of dissimilarity

- To decide which clusters should be *combined* (for agglomerative), or where a cluster should be *split* (for divisive), a **measure of dissimilarity** between sets of observations is required
- *The dissimilarity between two groups = measure of distance between the groups = defined in terms of distance between two observations*

Measures of distance between data points

- Numeric data: two observations $\vec{x}$ and $\vec{y}$

Name	Formula
Euclidean distance	$\|\vec{x}-\vec{y}\|_2 = \sqrt{(x_1-y_1)^2+\cdots+(x_n-y_n)^2}$
Squared Euclidean distance	$\|\vec{x}-\vec{y}\|_2^2 = (x_1-y_1)^2+\cdots+(x_n-y_n)^2$
Manhattan distance	$\|\vec{x}-\vec{y}\|_1 = \|x_1-y_1\|+\cdots+\|x_n-y_n\|$
Maximum distance	$\|\vec{x}-\vec{y}\|_\infty = \max\{\|x_1-y_1\|,\ldots,\|x_n-y_n\|\}$

Hierarchical Clustering methods V

- Non-numeric data: Levenshtein distance, Hamming Distance etc.
- Levenshtein :distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other.

kitten → sitten (substitution of "s" for "k")
sitten → sittin (substitution of "i" for "e")
sittin → sitting (insertion of"g" at the end)

Measures of distance between groups of data points

- Two groups A and B, x and y be arbitrary data points in A and B respectively
- $d(x, y)$: distance between x and y
- $d(A, B)$ the distance between the groups A and B
- The distance measures are:

Hierarchical Clustering methods VI

- **Maximum distance:** Maximum of all the distance measures between each pair data points x and y
- Known as *complete-linkage clustering* or *farthest neighbour clustering*

$$d(A, B) = max\{d(x, y) : x \in A, y \in B\}$$

- **Minimum distance:** Minimum of all the distance measures between each pair data points x and y
- Known as: *single-linkage clustering* or *nearest neighbour clustering*.

$$d(A, B) = min\{d(x, y) : x \in A, y \in B\}$$

- **Mean distance:** distance between the means of the two clusters A and B

$$d(A, B) = \mid m_A - m_B \mid$$

- **Average distance:** Average of all distance measures between each pair data points x and y

$$\frac{1}{\mid A \mid \mid B \mid} \dot{\sum_{x \in A, y \in B}} d(x, y)$$

Hierarchical Clustering methods VII

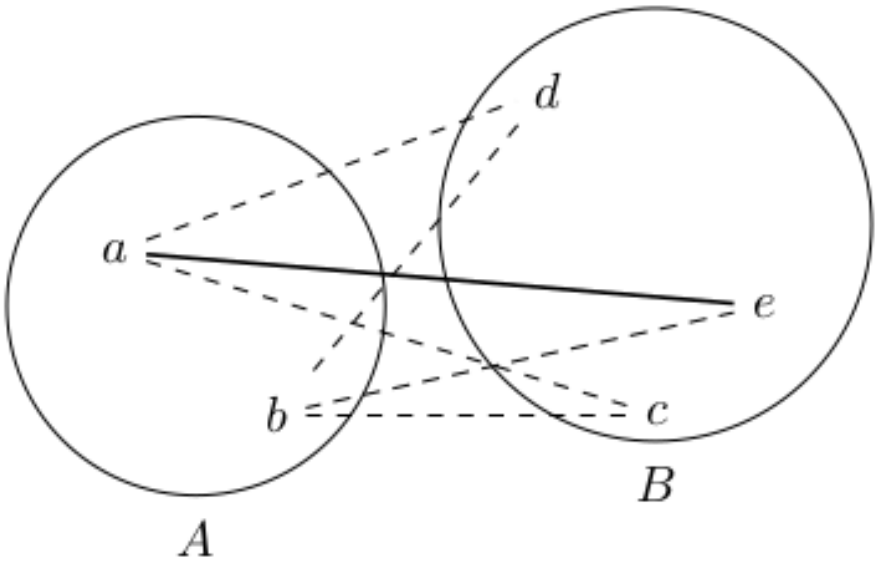


Figure: complete-linkage clustering

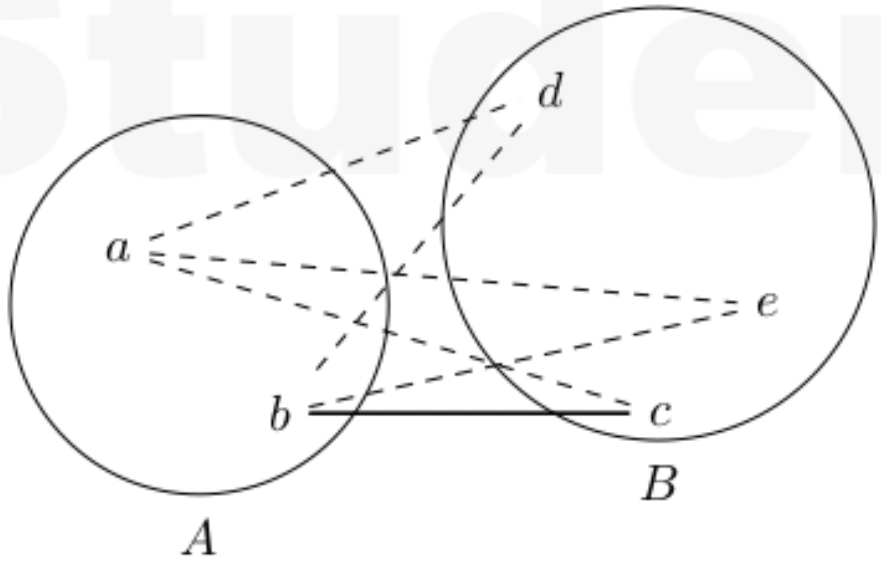
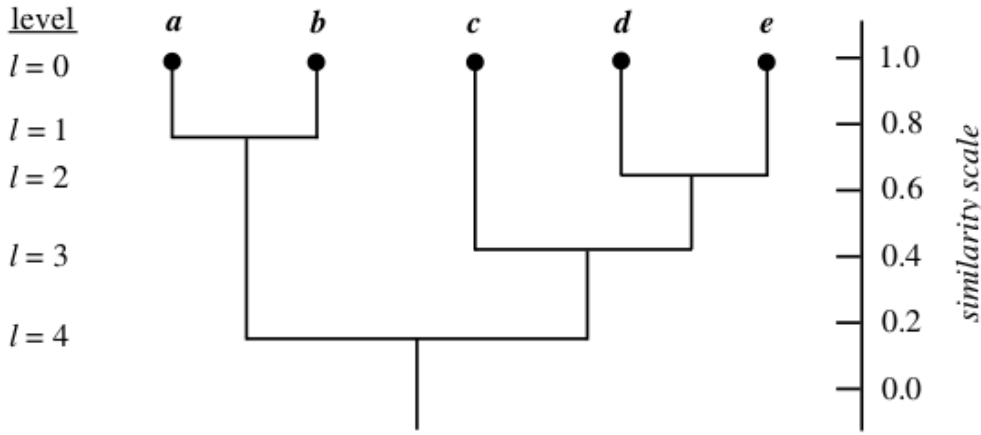


Figure: single-linkage clustering

Hierarchical Clustering methods VIII

Dendrogram

- It is **tree structure** commonly used to represent the process of hierarchical clustering
- It shows *how objects are grouped* together step by step
- Also use a vertical axis to show the similarity scale between clusters
- Eg; when the similarity of two groups of objects, $\{a, b\}$ and $\{c, d, e\}$, is roughly 0.16, they are merged together to form a single cluster.
- Fig: A dendrogram of the dataset $\{a, b, c, d, e\}$
- $l = 0$: shows the five objects as singleton clusters at level 0
- At $l = 1$, objects a and b are grouped together to form the first cluster



Hierarchical Clustering method:BIRCH I

BIRCH: Balanced Iterative Reducing and Clustering Using Hierarchies

- Designed for clustering a *large amount of numerical data*
- A *multiphase clustering algorithm* having two phase: microclustering phase, macroclustering phase
- Phase 1: microclustering phase
  - Performs a heirarchical clustering
  - Scan DB and incrementally construct a CF(clustering feature) tree which is a heirarchical data structure
  - The CF-tree is a multilevel compression of data that tries to preserve the inherent clustering structure of the data
  - Finally we get a CF tree having leaf nodes.
  - The leaf nodes represent clustering features of small tight clusters of the entire database
- Phase 2: macroclustering phase

Hierarchical Clustering method:BIRCH II

- Here, other arbitrary clustering methods such as iterative partitioning is used to cluster the leaf nodes of the CF tree
- This phase removes sparse clusters as outliers and groups dense clusters into larger ones.

- BIRCH works based on two concepts: **clustering feature** , **clustering feature tree (CF tree)**,
  - Theses two structures are used to summarize cluster representations.
  - They help the clustering method achieve good speed and scalability in large databases
  - Also make it effective for incremental and dynamic clustering of incoming objects.
- Clustering Feature (CF)
  - A three-dimensional vector summarizing information about clusters of objects
  - Given n $d-dimensional$ objects in a cluster, x_i, then the CF of the cluster is defined as:

Hierarchical Clustering method:BIRCH III

$$CF = \langle n, LS, SS \rangle \text{ or } CF = \langle n, \sum_{i=1}^{n} x_i, \sum_{i=1}^{n} {x_i}^2 \rangle$$

- Eg; Suppose we have three points, $(2,5), (3,2)$, and $(4,3)$, in a cluster, C_1. Then clustering feature of C_1 is:

$$CF_1 = \langle 3, (2+3+4, 5+2+3), (2^2+3^2+4^2, 5^2+2^2+3^2) \rangle$$

$$= \langle 3, (9,10), (29,38) \rangle$$

- Also Clustering features are additive. ie; if we merge two clusters C_1 and C_2 we get new cluster C_3 with $CF_3 = CF_1 + CF_2$
- We can see that A clustering feature is essentially a summary of the statistics for the given cluster
- With these statistics we can compute several measures related to each cluster. The measures are $centroid(x_0)$, $radius(R)$, $diameter(D)$

Hierarchical Clustering method:BIRCH IV

$$\boldsymbol{x_0} = \frac{\sum_{i=1}^{n} \boldsymbol{x_i}}{n} = \frac{LS}{n},$$

$$R = \sqrt{\frac{\sum_{i=1}^{n} (\boldsymbol{x_i} - \boldsymbol{x_0})^2}{n}} = \sqrt{\frac{nSS - 2LS^2 + nLS}{n^2}},$$

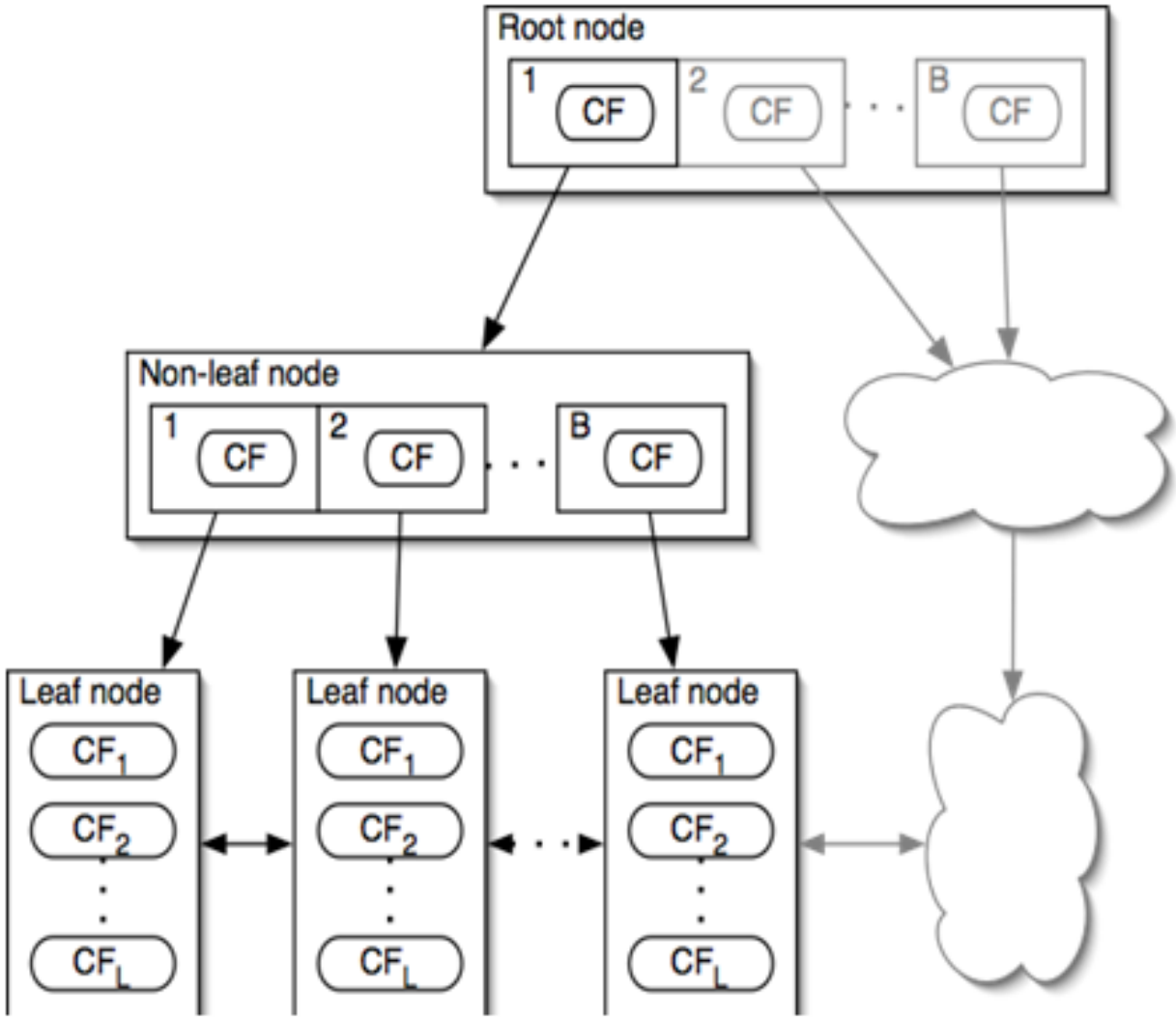
$$D = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n} (\boldsymbol{x_i} - \boldsymbol{x_j})^2}{n(n-1)}} = \sqrt{\frac{2nSS - 2LS^2}{n(n-1)}}.$$

- R : the average distance from member objects to the centroid
- D : is the average pairwise distance within a cluster.
- Both R and D reflect the tightness of the cluster around the centroid

Hierarchical Clustering method:BIRCH V

- CF tree
  - A height-balanced tree that stores the clustering features for a hierarchical clustering
  - Nonleaf nodes: summarize clustering information about their children
  - A CF tree has two parameters: $branching factor, B$, and $threshold, T$.
  - Branching factor specifies the maximum number of children per nonleaf node
  - The threshold parameter specifies the maximum diameter of subclusters stored at the leaf nodes of the tree.

Hierarchical Clustering method:BIRCH VI



Hierarchical Clustering method:BIRCH VII

- Working of BIRCH algorithm
  - Initially the rooted tree will be empty.
  - When it scans new $n-dimenstional$ data point(2,3), it summarize the data point as $CF_1 = \langle 1, LS, SS \rangle = \langle 1, (2,3), (4,9) \rangle$
  - When a new data points comes, the algorithm find the closest cluster to insert it by taking distance measures.
  - If it finds one, the cluster feature is updated as $CF_1 = \langle 2, LS, SS \rangle$. Otherwise if it is far away a new cluster CF_2 is created at the root.
  - You can see that by looking into cluster features we can have the centroid, radius and diameter so that the cluster for new data point can easily be computed.
  - This process continues till the no of clusters in the root reaches branching factor(B)
  - When root gets filled up split into two branches with two clusters at root with each cluster having summary of its children.
  - This is how the tree grows up.
  - Finally we get a lot of leafs having a number of cluster features in it.

Hierarchical Clustering method:BIRCH VIII

- These cluster features represent the tight clusters(clusters of small diameters) in our dataset.
- The size of the tight clusters is restricted by the *Threshold* value.
- That is to say, the diameter of each leaf entry has to be less than Threshold.
- Then we compute the centroids of each tight clusters for using them as representatives for furthur clustering.
- One we have these centroids, we start second phase, ie, we use any clustering algorithms to cluster the data points.
- Also note that the cluster features in leaf nodes does not cluster the points, instead they represent the features of tight clusters so that we can start clustering our data points using any of clustering algorithms.

Hierarchical Clustering method:BIRCH IX

The Algorithm

ALGORITHM 5.10

Input:

$D = \{t_1, t_2, \ldots, t_n\}$ //Set of elements

T // Threshold for CF tree construction

Output:

K //Set of clusters

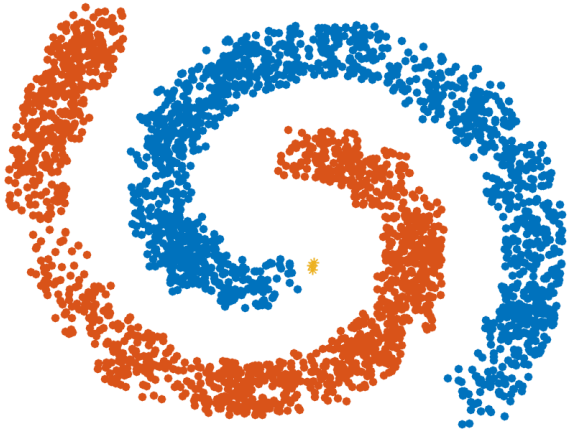
BIRCH clustering algorithm:

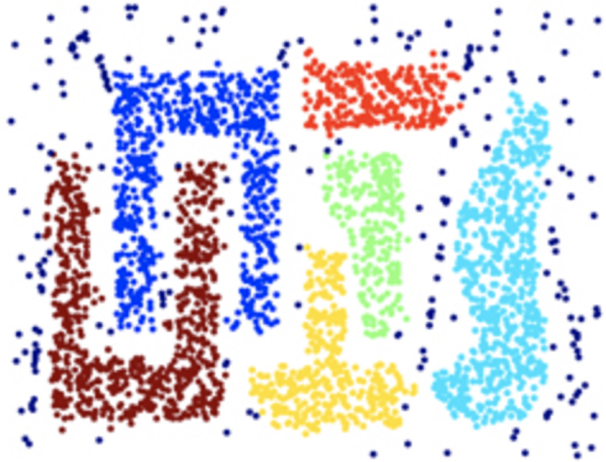
```
for each t_i ∈ D do
    determine correct leaf node for t_i insertion;
    if threshold condition is not violated, then
        add t_i to cluster and update CF triples;
    else
        if room to insert t_i, then
            insert t_i as single cluster and update CF triples;
        else
            split leaf node and redistribute CF features;
```

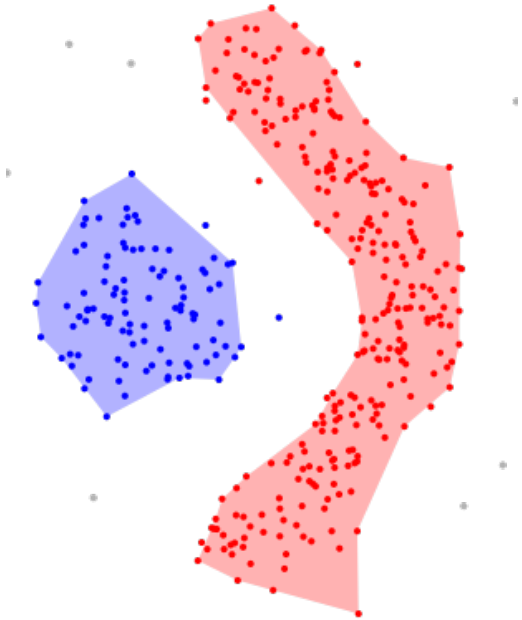
Density-Based Clustering methods: I

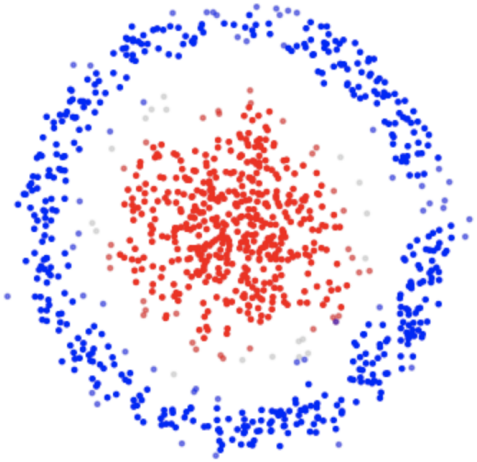
- Density-based clustering methods have been developed to discover clusters with arbitrary shape
- Represent clusters as dense regions of objects in the data space and regions of low density as noise
- In density-based clustering, clusters are defined as areas of higher density than the remainder of the data
- Objects in these sparse areas - that are required to separate clusters - are usually considered to be noise and border points.
- ie., Clusters are dense regions in the data space, separated by regions of the lower density of points.
- Eg; DBSCAN, OPTICS: an extension of DBSCAN

Density-Based Clustering methods: II









DBSCAN(Density-Based Spatial Clustering of Applications with Noise) I

Some terminologies:

- **Density**: density of an object p is the number of objects close to p
- ϵ-**neighborhood** of an object p is the space within a radius ϵ centered at p.
- A neighborhood is **dense (high density)** if it contains minimum number of points (m_0 or $MinPts$)
- **Core Object**: An object is a core object if the $\epsilon - neighborhood$ of the object contains at least m_0 objects
- **Border point** : a point p as a border point if $N_\epsilon(p)$ has fewer than m_0 points, but is in the $\epsilon - neighborhood$ of a core point q

DBSCAN(Density-Based Spatial Clustering of Applications with Noise) II

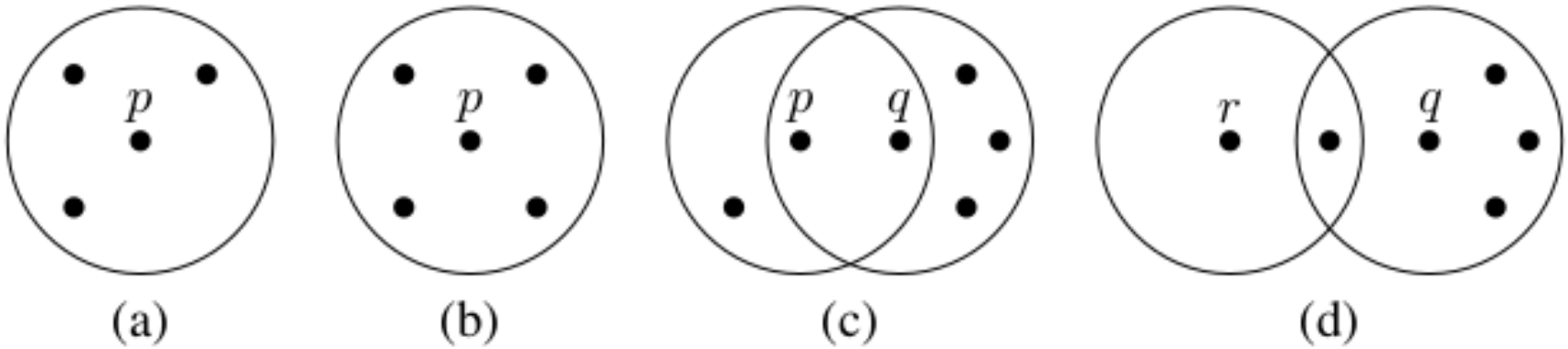


Figure: With $m_0 = 4$: (a) p a point of high density (b) p a core point (c) p a border point (d) r a noise point

DBSCAN(Density-Based Spatial Clustering of Applications with Noise) III

- **Directly density-reachable**: An object q is directly density-reachable from object p if p is a *core object* and q is in $N_\epsilon(p)$
- **Indirectly density-reachable**: An object q is indirectly density-reachable from an object p if there is a finite set of objects $p_1, ..., p_r$ such that p_1 is directly density-reachable from p, p_2 is directly density reachable from p_1 , etc., q is directly density-reachable form p_r.

DBSCAN(Density-Based Spatial Clustering of Applications with Noise) IV

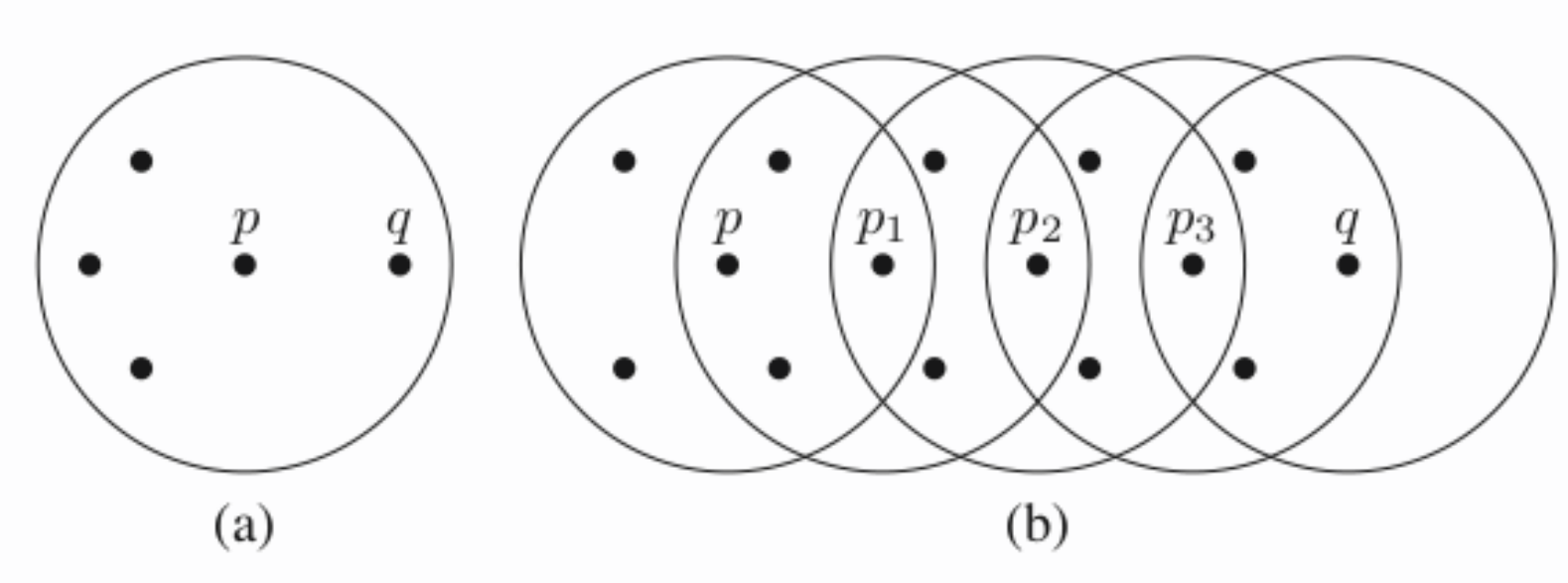


Figure: With $m_0 = 4$: (a) q is directly density-reachable from p (b) q is indirectly density-reachable from p

DBSCAN(Density-Based Spatial Clustering of Applications with Noise) V

DBSCAN algorithm

DBSCAN requires two parameters: ϵ (eps): specify the radius of a neighborhood and m_0: minimum number of points required to form a cluster.

(1) Start with an arbitrary starting point p that has not been visited.

(2) Extract the ϵ-neighborhood $N_\epsilon(p)$ of p.

$$N_\epsilon(p) = \{q : d(p,q) < \epsilon\}$$

(3) If the number of points in $N_\epsilon(p) < m_0$ (p *is border-point*)

- label point p as *noise*
- later this point can become the part of the cluster

(4) If the number of points in $N_\epsilon(p) \geq m_0$

- label point p is as *core point*
- mark p as *visited*

DBSCAN(Density-Based Spatial Clustering of Applications with Noise) VI

- Select a new *cluster-id* and mark all objects in $N_\epsilon(p)$ with this *cluster-id*.

(5) If a point q is found to be a part of the cluster(direct density reachable) then its $\epsilon - neighborhood$ is also the part of the cluster(indirect density reachable).

- Repeat from Step 2 for all $\epsilon - neighborhood$ until all points in the cluster are determined

(6) Retrieve new *unvisited* point and process it, which leads to the discovery of a further *cluster* or *noise*.

(7) Repeat the process until all points are marked as *visited*.

DBSCAN(Density-Based Spatial Clustering of Applications with Noise) VII

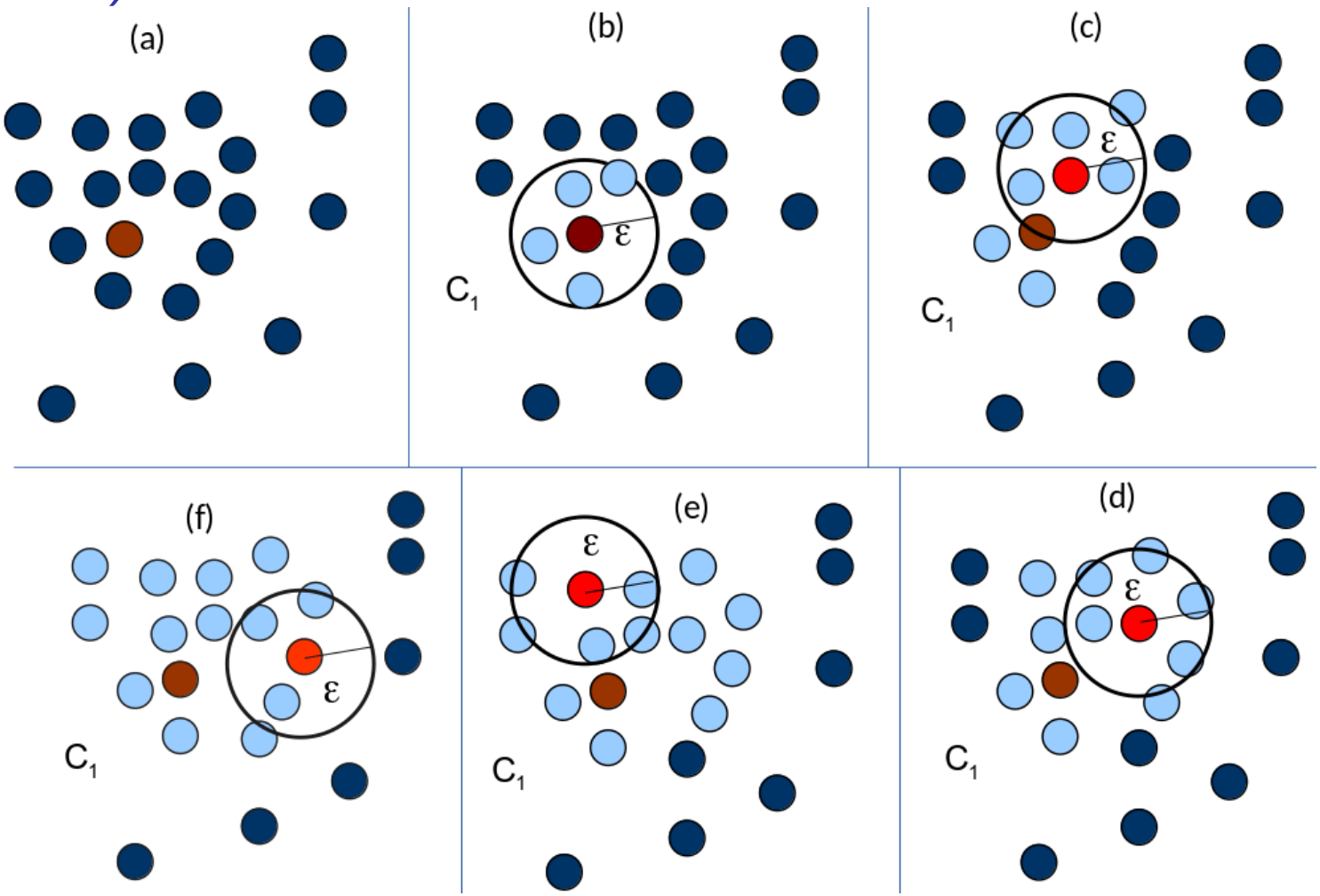


Figure: DBSCAN Example

DBSCAN(Density-Based Spatial Clustering of Applications with Noise) VIII

Advantages

- Can discover arbitrarily shaped clusters
- Find clusters completely surrounded by different clusters
- Robust towards outlier detection(noise)
- Insensitive to ordering of the arbitrary points
- No need to give the number of clusters required

Disadvantages

- Sensitive to clustering parameters ϵ and $MinPts$
- Fails to identify clusters with varying densities and sparse datasets
- Sampling affetcs density measure
- Sometimes has issues seperating nearby clusters properly
- It relies on reachability distance

OPTICS: Ordering Points to Identify the Clustering Structure I

- OPTICS algorithm is an extension of DBSCAN algorithm
- In the case DBSCAN algorithm
  - It cluster objects when given input parameters: $\epsilon, MinPts$
  - Users has the responsibility of selecting parameter values that will lead to the discovery of acceptable clusters.
  - Such parameter settings are usually empirically set and difficult to determine.
  - For datasets with very skewed distributions(varying densities), Using a single set of global density parameters($\epsilon, MinPts$) can not characterize the intrinsic clustering structure well.
- OPTICS was proposed to to overcome the difficulty in using one set of global parameters in clustering analysis
- OPTICS does not explicitly produce a data set clustering, Instead, it outputs a *cluster ordering*.

OPTICS: Ordering Points to Identify the Clustering Structure II

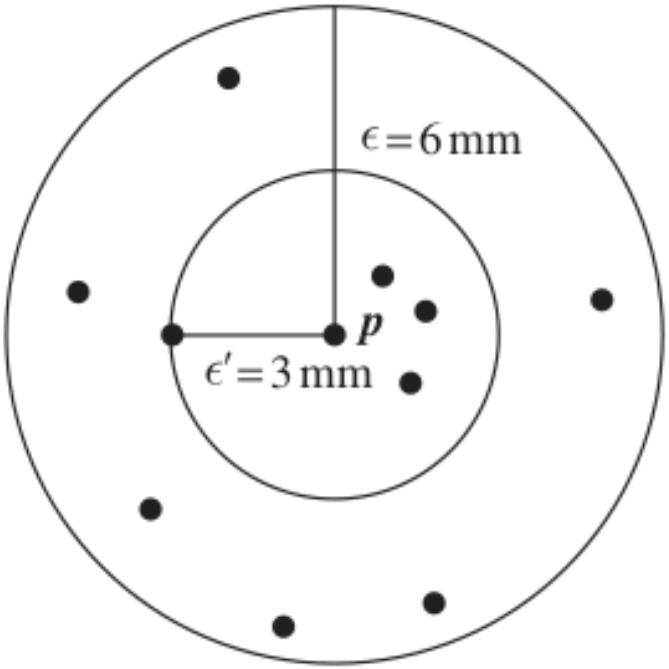
- Cluster ordering is a linear list of all objects under analysis and represents the density-based clustering structure of the data.
- OPTICS does not require the user to provide a specific density threshold.
- Instead can provide density-based clustering obtained from a wide range of parameter settings
- Also Cluster Ordering helps to
  - to construct the different clusterings simultaneously
  - to extract basic clustering information (e.g., cluster centers, or arbitrary-shaped clusters)
  - derive the intrinsic clustering structure
  - provide a visualization of the clustering

OPTICS: Ordering Points to Identify the Clustering Structure III

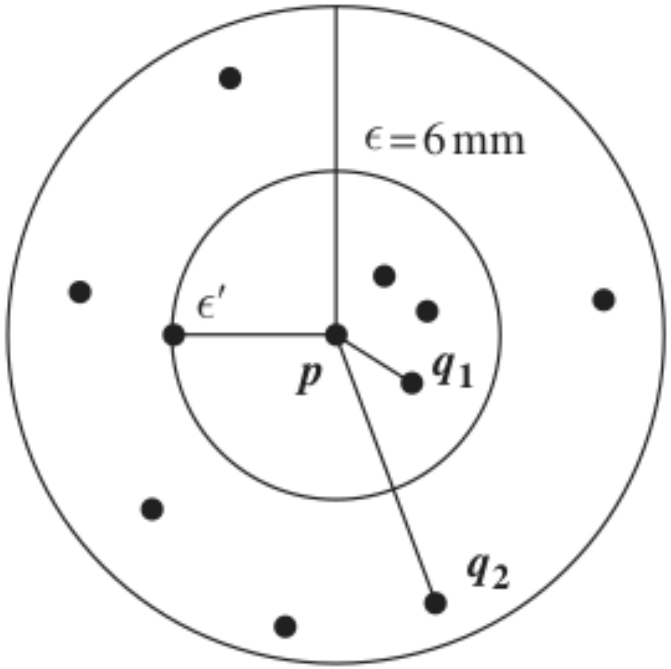
OPTICS introduce two important concepts

- *Core-distance*:
  - Core-distance of p is the smallest radius ϵ' such that the $\epsilon' - neighborhood$ of p has at least $MinPts$ objects.
  - ie; ϵ' is the minimum distance threshold that makes p a core object
  - If p not a core object, then, $core\_distance = UNDEFINED$
- *Reachability-distance* of q from core-object p:
  - If q is $\epsilon' - neighborhood$ of p then $r\_dist = core\_distance$ of p, (ie; if $dist(p,q) < core\_distance$ of p)
  - If $dist(p,q) > core\_distance$ of p, then $r\_dist = dist(p,q)$, where $dist(p,q)$ is the Euclidean distance from p to q
  - If p not a core object, then, $r\_dist = UNDEFINED$

OPTICS: Ordering Points to Identify the Clustering Structure IV



Core-distance of p



Reachability-distance $(p, q_1) = \epsilon' = 3\,\text{mm}$
Reachability-distance $(p, q_2) = dist\,(p, q_2)$

Figure: Suppose that $\epsilon = 6mm$ and $MinPts = 5$

OPTICS: Ordering Points to Identify the Clustering Structure V

OPTICS: Algorithm

OPTICS maintains a list called $OrderSeeds$ to generate the output ordering.

(1) Select a random object $p \in D$

(2) Retrieve $N_\epsilon(p)$

(3) Set the parameters for p

- Set $p.visited = TRUE$
- $p.r\_dist = UNDEFINED$
- if p is a core-object find $p.core\_dist$, otherwise $p.core\_dist = UNDEFINED$

- If p is not a core object,
  - OPTICS simply picks next object in the $OrderSeeds$ list (or the database D if $OrderSeeds$ is empty).

(4) If p is core-object, then for each neighbor q in $N_\epsilon(p)$

OPTICS: Ordering Points to Identify the Clustering Structure VI

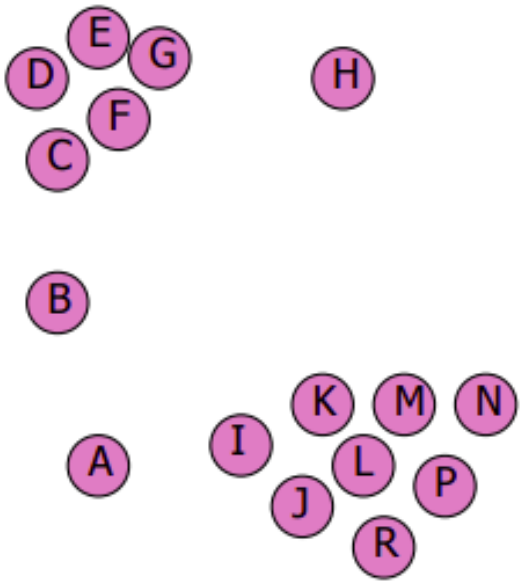
- find reachability distance($q.r\_dist$) from p
- add q into $OrderSeeds$ inserts q into OrderSeeds if q has not yet been processed.
- Note that the $OrderSeeds$ maintains object the order of reachability distance.
- If the object q is already in $OrderSeeds$ then update the reachability distance with the minimun value.

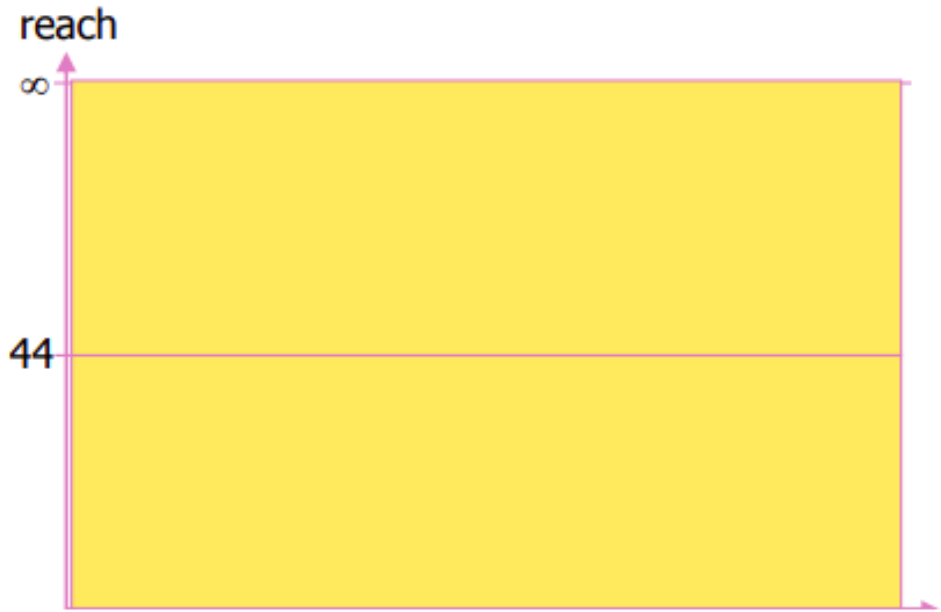
(5) Repeat from steps (2) until the input is fully consumed and $OrderSeeds$ is empty

OPTICS: Ordering Points to Identify the Clustering Structure VII

OPTICS: Example

- Example Database (2-dimensional, 16 points)
- $\varepsilon = 44$, $MinPts = 3$





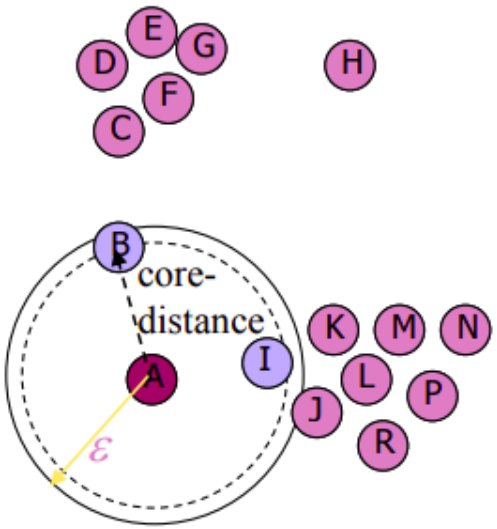
OPTICS: Ordering Points to Identify the Clustering Structure VIII

(1) Picks A, A is Core Object, Find $N_\epsilon(A) = \{B, I\}$, $A.r\_dist = \infty$

(2) For each neighbor in $N_\epsilon(A)$ find reachability and insert to $OrderSeeds$

$$OrderSeeds : (B, 40)(I, 40)$$

- $\varepsilon = 44$, $MinPts = 3$



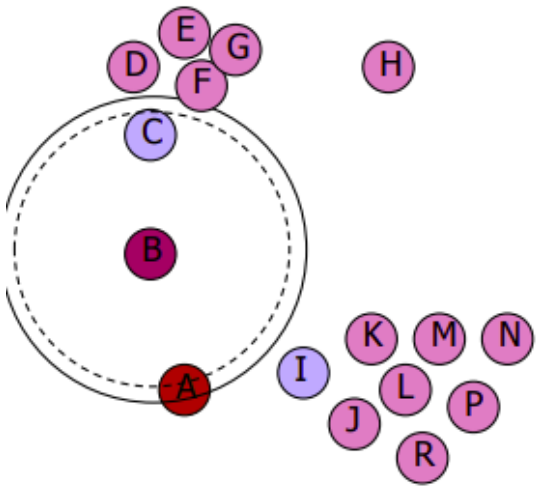
OPTICS: Ordering Points to Identify the Clustering Structure IX

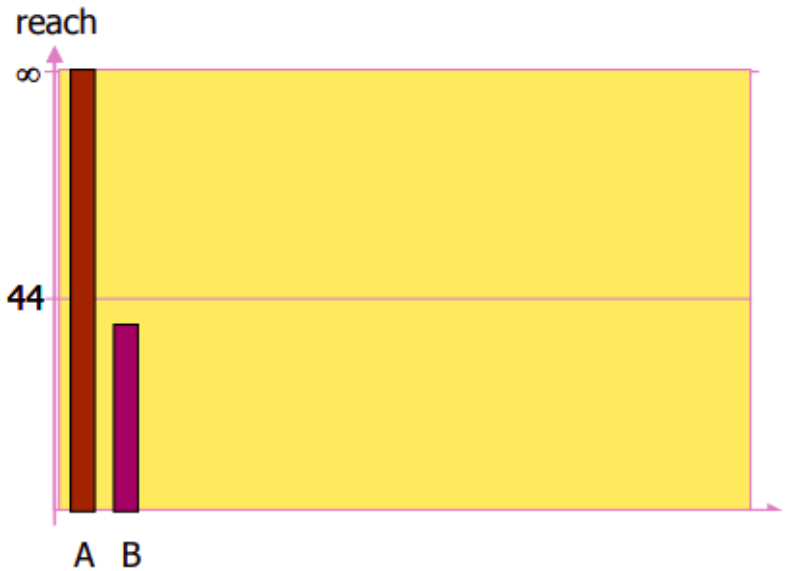
(3) Took next item from $OrderSeeds$, Picks B, B is Core Object, Find $N_\epsilon(B) = \{A, C\}$

(4) For each neighbor in $N_\epsilon(B)$ find reachability and insert to $OrderSeeds$,A is visited

$$OrderSeeds : (I, 40)(C, 40)$$

- $\varepsilon = 44$, $MinPts = 3$





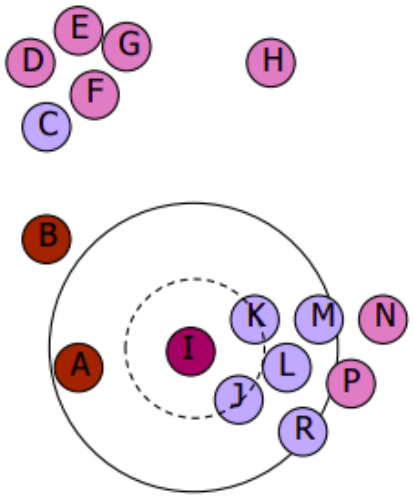
OPTICS: Ordering Points to Identify the Clustering Structure X

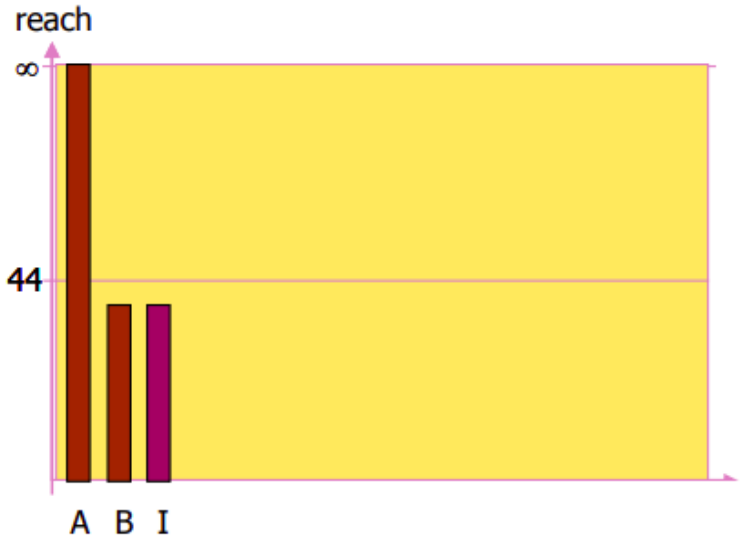
(5) Took next item from $OrderSeeds$, Picks I, I is Core Object, Find $N_\epsilon(I) = \{A, J, K, L, M, R\}$

(6) For each neighbor in $N_\epsilon(I)$ find reachability and insert to $OrderSeeds$, A is visited

$$OrderSeeds : (J, 20)(K, 20)(L, 31)(C, 40)(M, 40)(R, 43)$$

- $\varepsilon = 44$, $MinPts = 3$



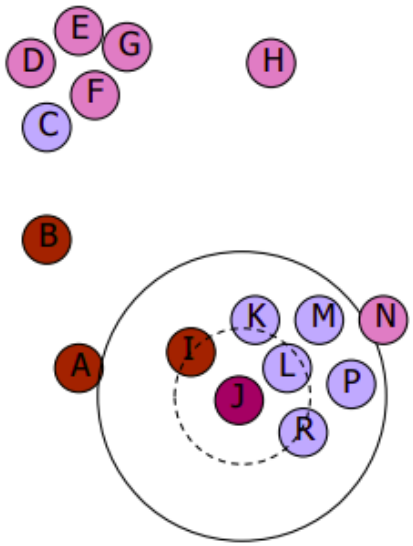


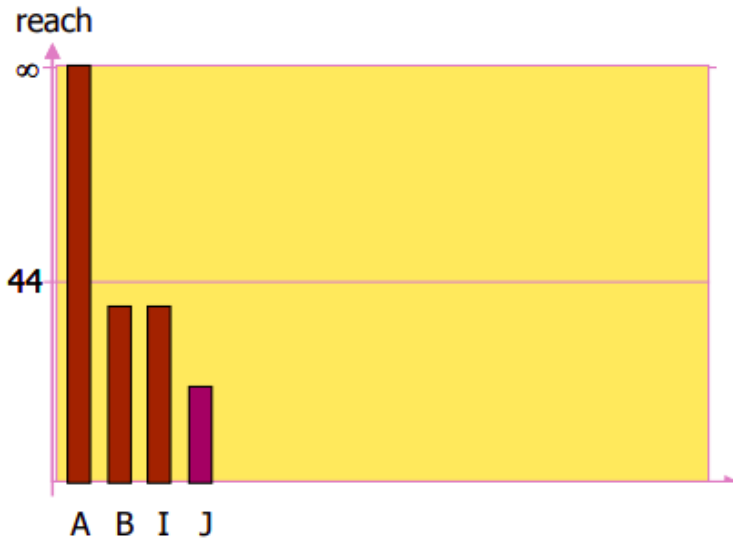
OPTICS: Ordering Points to Identify the Clustering Structure XI

(7) Next is J which is a Core Object, Find $N_\epsilon(J) = \{I, K, L, M, R, P\}$

(8) For each neighbor in $N_\epsilon(J)$ find reachability and insert to $OrderSeeds$, I is visited, L, K, R, M already in $OrderSeeds$, Update reachability distances

$$OrderSeeds : (L, 19)(K, 20)(R, 21)(M, 30)(P, 31)(C, 40)$$



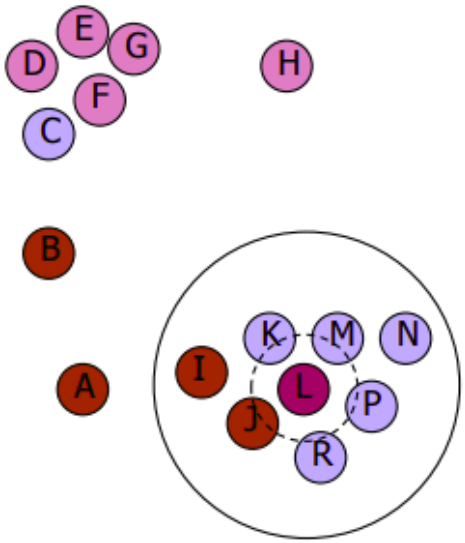


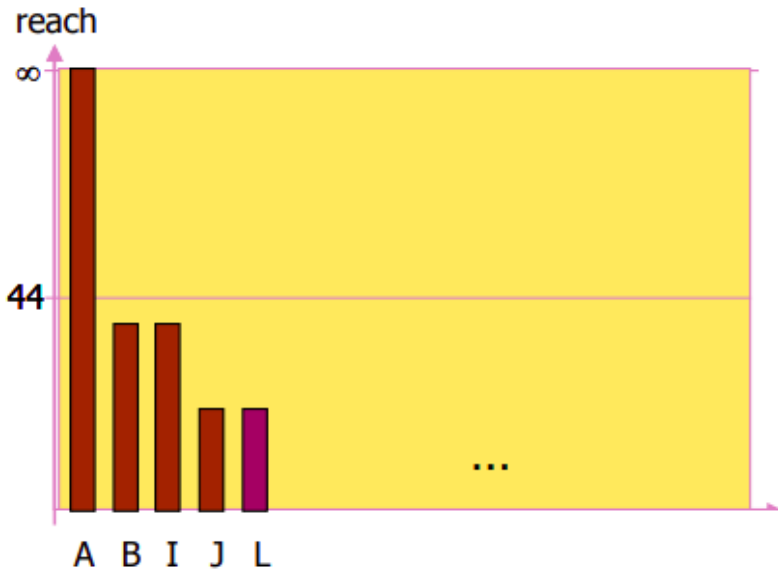
OPTICS: Ordering Points to Identify the Clustering Structure XII

(9) Picks L, which is Core Object, Find $N_\epsilon(L) = \{I, J, K, M, R, P, N\}$

(10) For each neighbor in $N_\epsilon(L)$ find reachability and insert to $OrderSeeds$, I, J is visited, K, M, P, R already in $OrderSeeds$, Update reachability distances

$$OrderSeeds : (M, 18)(K, 18)(R, 20)(P, 21)(N, 35)(C, 40)$$

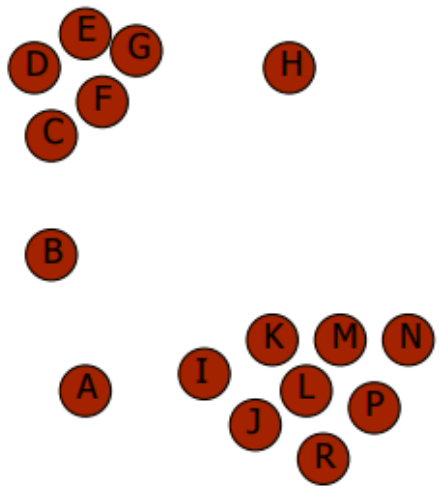


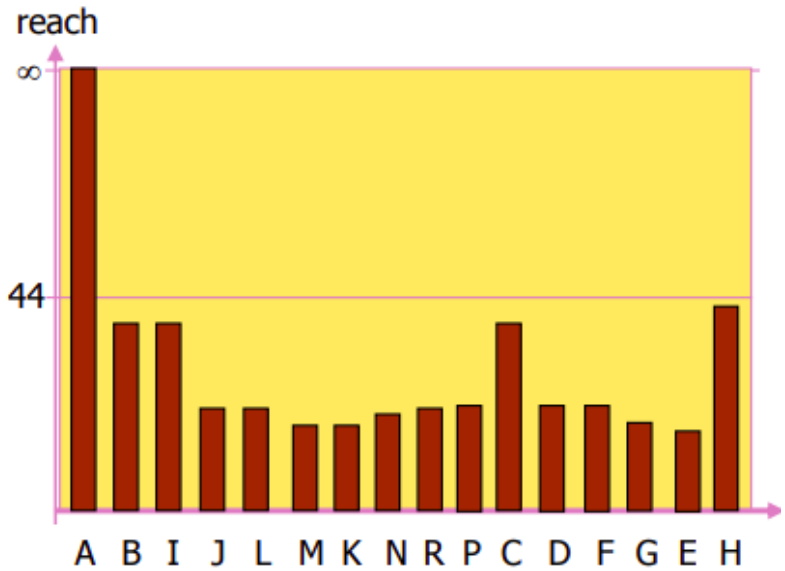


OPTICS: Ordering Points to Identify the Clustering Structure XIII

(11) Proceeding like this we, finally visit all the points and get the reachability plot as follows

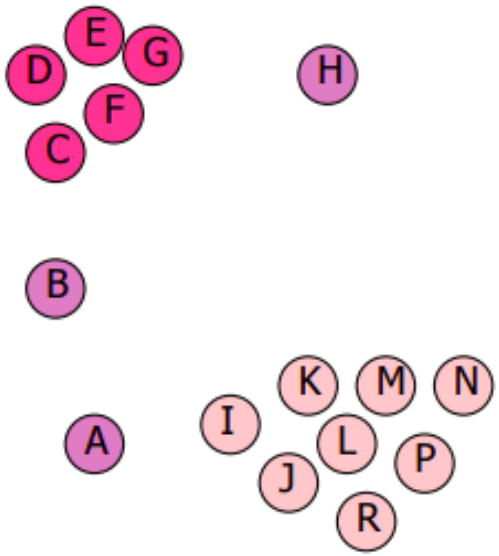
$$OrderSeeds : empty$$

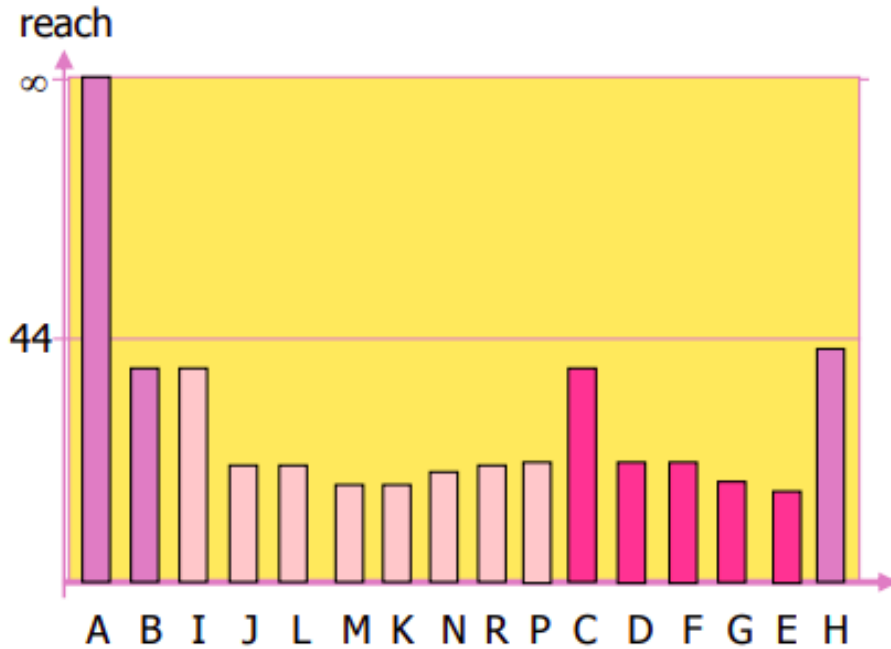




OPTICS: Ordering Points to Identify the Clustering Structure XIV

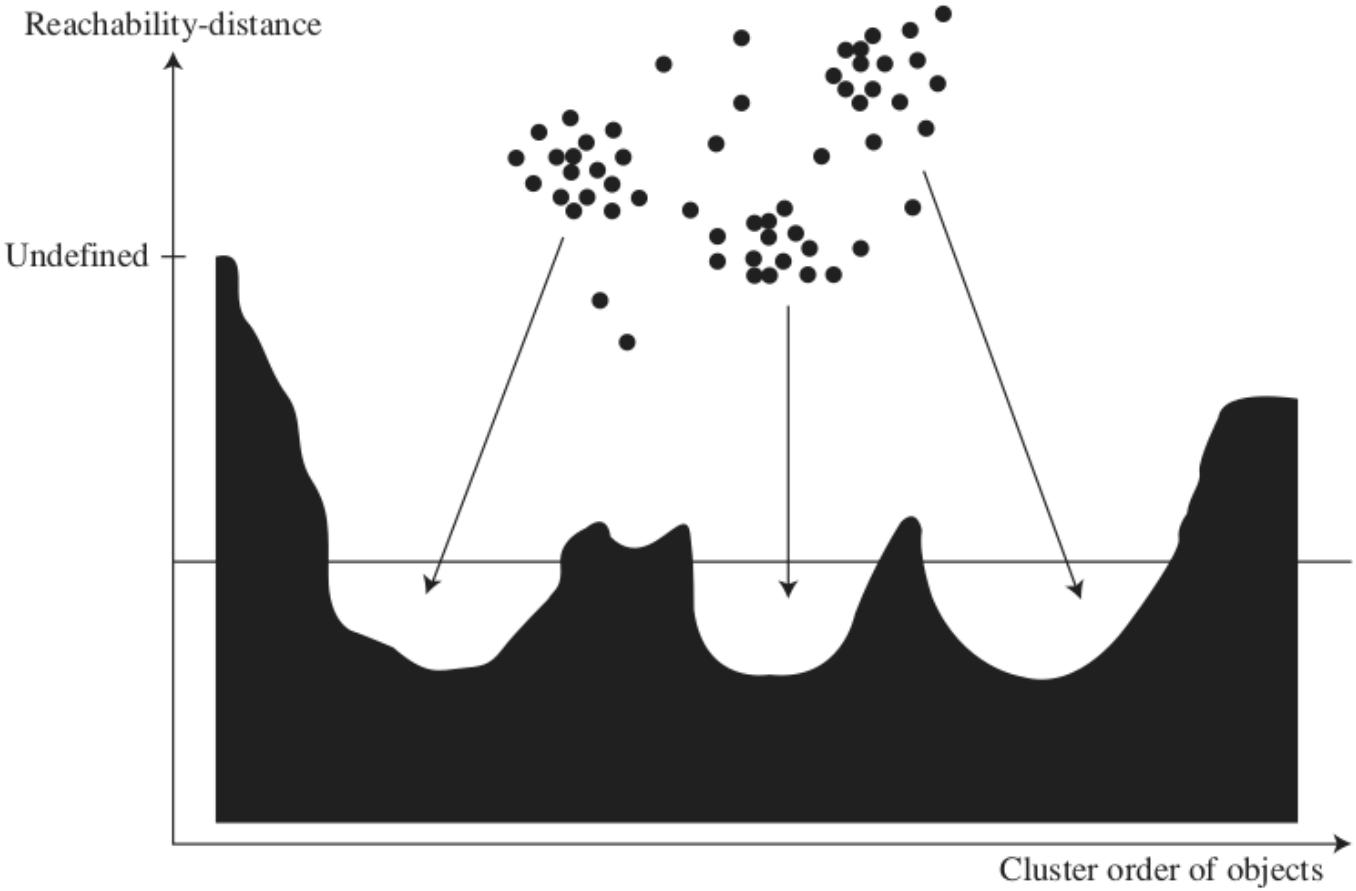
(12) The clusters can be selected as follows





OPTICS: Ordering Points to Identify the Clustering Structure XV

Example: Reachability plot another dataset result of Cluster Ordering in OPTICS



Web Mining I

- The World Wide Web, commonly known as the Web
- An information system where documents and other web resources are identified by Uniform Resource Locators(URL).
- These documents are interlinked using hypertext links and are accessible via the Internet.
- Web mining is *mining of data related to the World Wide Web*.

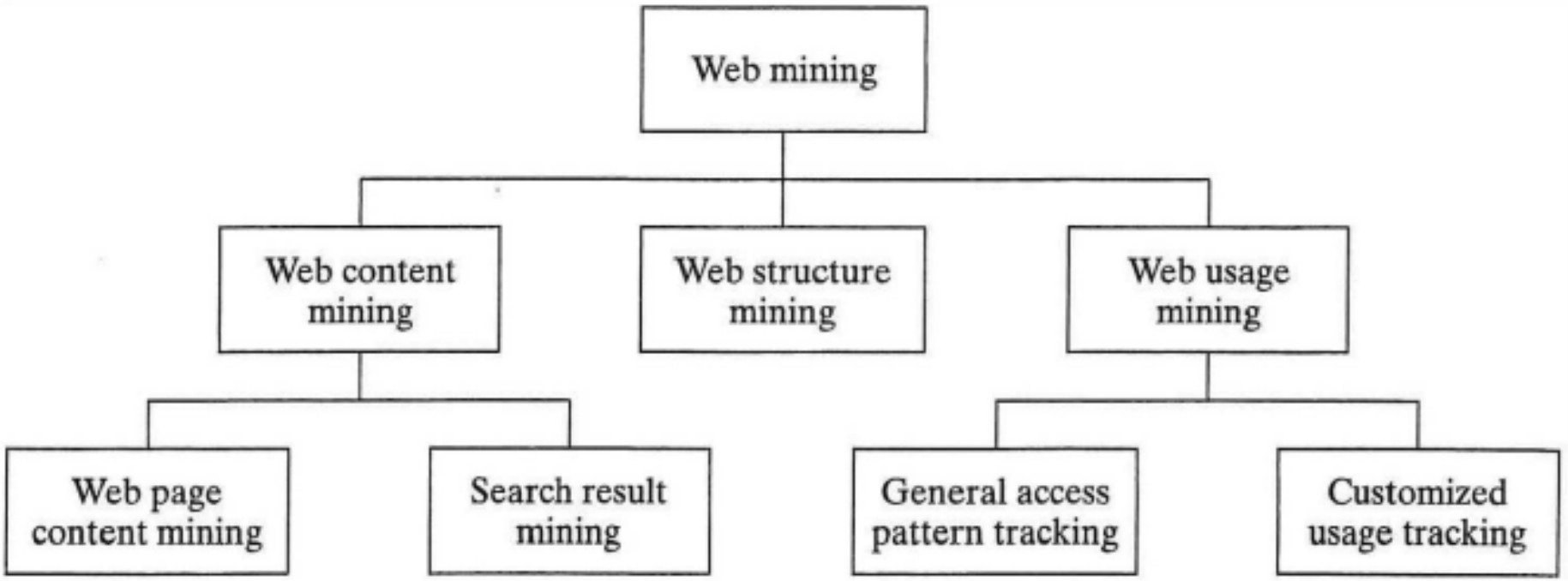
Web Mining II

Classification of Web Data

- *Content* of actual Web pages.
- *Intrapage structure* includes the HTML or XML code for the page.
- *Interpage structure* is the actual linkage structure between Web pages.
- *Usage data* that describe how Web pages are accessed by visitors.
- *User profiles* include demographic and registration information obtained about users. (Eg; database, information found in cookies)

Web Mining III

Classification of Web mining tasks



- Web content mining:
  - examines the *content of Web pages* as well as *results of Web searching*.

Web Mining IV

- Web page content mining is the traditional searching of Web pages via content
- The content includes *text, audio, video, graphics, structured records* (within web pages or linked across web pages) etc.
- Search result mining is a further *search of pages* found from a previous search

- Web structure mining:
  - Mines the information from the actual *organization of pages* on the Web
  - This include the *intrapage structure* mining and *interpage structure* mining
  - The intrapage structure includes links within the page as well as the code (HTML, XML) for the page.
  - Interpage structure is the actual linkage structure between Web pages.
- Web usage mining:
  - process of extracting useful information (e.g., user click streams) from *logs of web access*.

Web Mining V

- Logs includes the *web server logs, application server logs* etc.
- General access pattern tracking is a type of usage mining that looks at a history of Web pages visited
- It finds patterns related to general or particular groups of users; understands users' *search patterns, trends, and associations*; and *predicts* what users are looking for on the Internet.

Web Mining VI

Applications of Web mining

- Targeted advertising:
  - Used for direct business marketing or advertising to the most beneficial subset of the total population
  - display advertising at Web sites visited by persons that fit into a business target demographic area.
- used to predict user behavior
- helps to improve the power of web search engine by classifying the web documents and identifying the web pages.
- used for Web Searching e.g., Google, Yahoo etc and Vertical Searching e.g., FatLens, Become etc.

Web Content Mining I

- It analyze the *web content* such as text, multimedia data, and structured data (within web pages or linked across web pages).
- Web content mining can be thought of as *extension* of the work performed by *basic search engines*
- Search engines use basic *IR(Information Retrieval) technologies* such as keyword based indexing
- Web content mining goes beyond this basic IR technology
- It use techniques such as *concept hierarchies and synonyms, user profiles* etc.
- Web content mining activities have centered around techniques to *summarize the information* found.
- The web content mining helps
  - to understand the content of web pages
  - provide scalable and informative keyword-based page indexing

Web Content Mining II

- entity/concept resolution
- web page relevance and ranking
- web page content summaries
- other valuable information related to web search and analysis.

- Web content mining divided into **agent-based approaches** and **database approaches**
- Agent-based approach
  - Have software systems (agents) that perform the content mining
  - Search engines belong to this class
  - Agents act as intelligent search agents, information filtering, and personalized Web agents.
  - Intelligent search agents use user profiles or knowledge concerning specified domains
  - Information filtering utilizes IR techniques, knowledge of the link structures, and other approaches to retrieve and categorize documents.

Web Content Mining III

- Personalized Web agents use information about user preferences to direct their search
- Examples: Web Crawlers, Harvest System

- Database approaches:
  - View the Web data as belonging to a database
  - Approaches that view the Web as a multilevel database, and
  - Uses many query languages that target the Web.
  - Examples: Virtual Web View

Web Content Mining IV

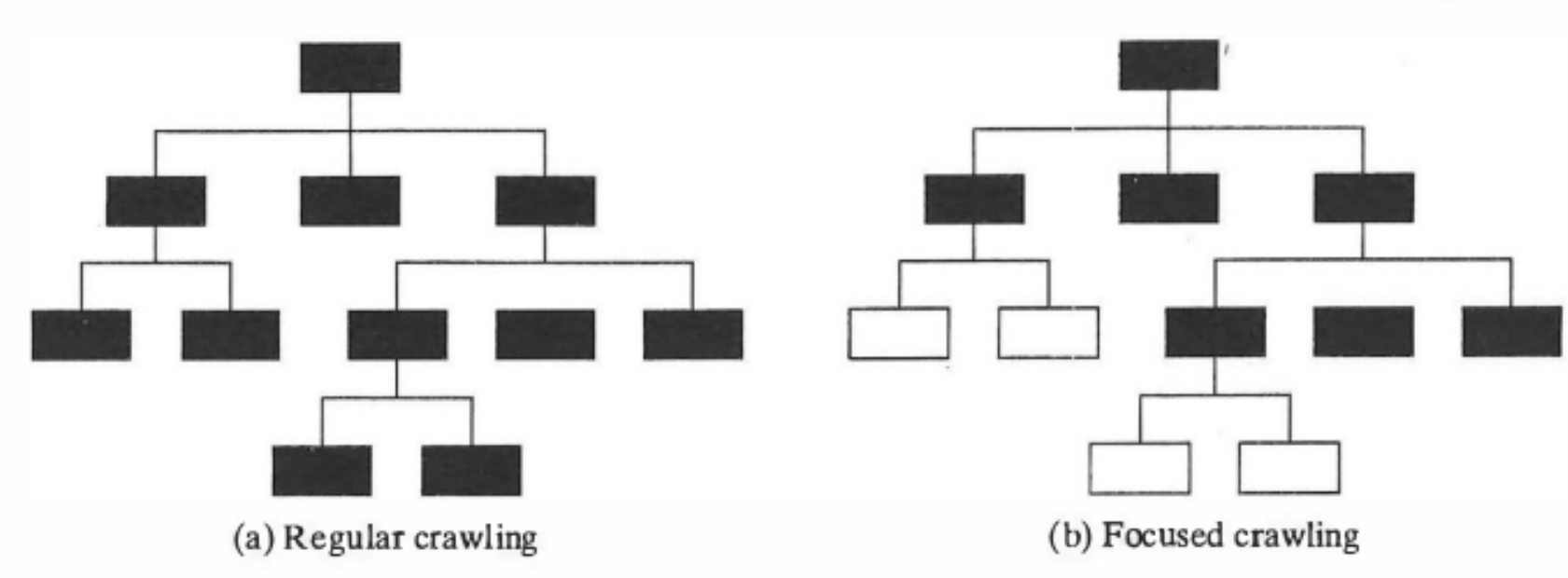
Web Crawlers

- It is an agent-based approach of web content mining
- A *robot* (or *spider* or *crawler*) is a program that traverses the hypertext structure in the Web.
- The crawlers starts with a page (or set of pages) referred to as the *seed* URLs.
- By starting at one page, all links from it are recorded and saved in a queue
- These new pages are in turn searched and their links are saved.
- ie; Crawlers are used to facilitate the creation of indices used by search engines
- There are different types of crawlers based on index update they use: periodic crawler, incremental crawler
- periodic crawler

Web Content Mining V

  - Visit a certain number of pages and then stop, build an index, and replace the existing index.
- incremental crawler
  - Selectively searches the Web and only updates the index incrementally.
- Crawlers can be further classified based on searching pattern: focused crawler, regular crawling
- Focused crawler:
  - A focused crawler visits pages related to topics of interest only
  - If it is determined that a page is not relevant or its links should not be followed, then the entire set of possible pages underneath it are pruned and not visited.
- Regular crawling:
  - A regular crawler visits all pages that the crawler indexes
  - It visits all the pages whether it is relevant or not.

Web Content Mining VI



(a) Regular crawling

(b) Focused crawling

Web Content Mining VII

Harvest System

- The Harvest system is based on the use of caching, indexing, and crawling.
- The harvest is a set of tools that facilitate gathering of information from diverse sources.
- It has two parts: a gatherer and a broker
- The Gatherer
  - The gatherer obtains information for indexing from an Internet service provider
  - It has three sub-components: the enumerator and the summariser (essence), gather daemon
  - Enumerator: fetches documents, extracts a list of all of the links that they contain, and then passes the document on to essence (the summariser)
  - Essence (the summariser): create a summary of the contents in the documents.

Web Content Mining VIII

- Gather daemon: makes the generated database available for Brokers to collect in either a plaintext or a compressed format

- The Broker
  - The broker provides the index and query interface.
  - It provides the search interface to the information collected by the Gatherer.
  - Brokers may interface directly with gatherers or may go through other brokers to get to the gatherers.
  - It is made of a number of components - a daemon, an indexing system, and a few CGIs.
  - Daemon: fetches data from gather daemons and from other Brokers, storing it in a directory structure on the local disk.
  - Indexing system: A seperate, customisable, indexing engine to index and search the data
  - The CGIs: provide an interface between a web form to perform the query, and the broker daemon.

Web Content Mining IX

Virtual Web View

- Virtual Web View (VWV) is a view of the MLDB(multiple layered database)
- MLDB is an approach to handling the large amounts of somewhat unstructured data on the Web
- It is a multiple layered, massive and distributed database created on top of the data in the Web (or a portion thereof).
- The MLDB provides an abstracted and condensed view of a portion of the Web
- Each layer of MLDB is more generalized than the layer beneath it
- MLDB does not require the use of spiders to build indexing
- Instead, the Web servers (masters, administrators) themselves send their indices (or changes to indices) to the MLDB.
- First layer in MLDB is created as follows:

Web Content Mining X

- Convert Web documents to XML using Translation tools
- Extract the desired information from the Web pages and insert it into the first layer using extraction tools

- Constructing higher levels:
  - The higher levels of the database become less distributed and more summarized as they move up the hierarchy
  - Generalization tools and concept hierarchies are used for generalization process

Web Structure Mining I

- Creates a model of the organization of Web or a portion thereof
- Mines the information from the actual organization of pages on the Web
- This include the intrapage structure mining and interpage structure mining
- The intrapage structure includes links within the page as well as the code (HTML, XML) for the page.
- Interpage structure is the actual linkage structure between Web pages.
- The model can be used to classify Web pages or to create similarity measures between documents.
- There are several techniques used for web structure mining: *PageRank, Clever*

Web Structure Mining II

PageRank

- This technique designed to both increase the effectiveness of search engines and improve their efficiency
- PageRank measures the importance of a page and help to prioritize pages returned from a traditional search engine using keyword searching.
- Calculating PageRank value
  - The PageRank value for a page is calculated based on the number backlinks to a page.
  - A backlinks are links pointing to a page rather than pointing out from a page
  - The PageRank is not simply a count of the number of backlinks also give weights backlinks coming from important pages.
  - Given a page p, The PageRank of p is:

$$PR(p) = c \dot{\sum_{q \in B_p}} \frac{PR(q)}{|F_q|}$$

Web Structure Mining III

- B_p : set of pages that point to p(backlinks),
- F_q : set of links out of q.
- c : a constant between 0 and 1,used for normalization
- rank sink problem: when a *cyclic reference* occurs (page A points to page B and page B points to page A), the PR value for these pages increases
- Solution: adding an additional term to the formula

$$PR'(p) = c \sum_{q \in B_p} \frac{PR(q)}{|F_q|} + cE(v)$$

- c is maximized
- $E(v)$ is a vector that adds an artificial link, which is a random surfer who periodically decides to stop following links and jumps to a new page.

Web Structure Mining IV

Clever

- Clever aimed at finding both *authoritative pages* and *hubs*
- *authoritative pages*: pages which authorized by authors as the "best source" for the requested information
- *hubs*: a page that contains links to authoritative pages
- The Clever system identifies best hubs and authoritative pages by creating weights.
- The pages which are seems to be higher quality than others are often referred to as being the most authoritative.
- A page may be extremely relevant, but if it contains factual errors, users certainly do not want to retrieve it.
- The issue of authority usually does not surface in traditional IR.
- Hyperlink-induced topic search (HITS) algorithm used to find hubs and authoritative pages

Web Structure Mining V

- The HITS technique contains two components:
  - Based on a given set of keywords (found in a query), a set of relevant pages (perhaps in the thousands) is found.
  - Hub and authority measures are associated with these pages. Pages with the highest values are returned.

Web Usage Mining I

- Web usage mining performs mining on Web usage data, or Web logs
- A Web log is a listing of page reference data
- Logs includes the web server logs, application server logs etc.
- Also called *clickstream data* because each entry corresponds to a mouse click.
- Web logs can be examined from either a client perspective or a server perspective.
- **Server perspective evaluation**:
  - Uncovers information about the sites where the service resides.
  - It can be used to improve the design of the sites.
- **Client perspective evaluation:**
  - Evaluates a client's sequence of clicks, thus information about a user (or group of users) is detected.
  - Used to perform prefetching and caching of pages

Web Usage Mining II

- Example of Web Usage Mining:
  - A webmaster at ABC Corp. learns high percentage of users have the following pattern of reference to pages: (A, B, A, C)
  - This means that a user accesses page A, then page B, then back to page A, and finally to page C
  - Based on this observation, he determines that a link is needed directly to page C from page B. He then adds this link.
- Web usage mining actually consists of three separate types of activities: Preprocessing,Pattern discovery, Pattern analysis
  - **Preprocessing**: focus on reformatting the Web log data before processing.
  - **Pattern discovery**: activities look to find hidden patterns within the log data.
  - **Pattern analysis**: process of looking at and interpreting the results of the discovery activities.

Web Usage Mining III

Preprocessing

- Web usage log probably is not in a format that is usable by mining applications
- Any data to be used in a mining application, the data may need to be reformatted and cleansed
- The preprocessing phase include cleansing, user identification session identification, path completion, and formatting
- A log is a set of triples: $\{\langle u_1, p_1, t_1\rangle, ...\langle u_n, p_n, t_n\rangle\}$
  - where, $u_i \in U$, the set of users
  - $p_i \in P$ be a set of literals, called *pages* or *clicks*,
  - t_i is a timestamp.
  - ie; a standard log data consist of the following: *source site(identified by User ID), destination site(Identified by page ID)*, and *time stamp*
- The following preprocessing activities can be done on log data:

Web Usage Mining IV

- Page addresses may be changed into unique page identifications (such as alphabetic characters) for security or privacy reasons
- Data may be cleansed by removing any irrelevant information. Eg; the log entries with figures (gif, jpg, etc.) can be removed.
- Data from the log may be grouped together to better understand the patterns of page references from each user (source site).
- References to the same site may be identified and examined to better understand who visits this page.
- The web servers usually divide the log records into sessions. the preprocesing combine all session records from the same source site that occur within a time period.
  - A session would be identified by a user logging into a computer, performing work, and then logging off.
  - A session S is an ordered list of pages accessed by a user u_i
  - i.e., $S = (\langle p_1, t_1 \rangle, \langle p_2, t_2 \rangle, ..., \langle p_n, t_n \rangle)$
  - where $\{\langle u_i, p_1, t_1 \rangle, ...\langle u_i, p_n, t_n \rangle\} \subseteq L$, where L is the log.

Web Usage Mining V

- Issues associated with the preprocessing activities
  - User Tracking is difficult
    - User Tracking is complicated by the use of proxy servers, client side caching, and corporate firewalls.
    - Source URL or IP address that indicates the source of the request, may not always be accurate in determining the source location of the visitor
  - Same user may use Multiple ISPs
    - Users who access the Internet through an Internet service provider (I SP) will all have the source location of that provider.
    - It is not unique to the individual.
  - Complications in identifying sequence of pages accessed
    - Identifying the actual sequence of pages accessed by a user is complicated by the use of client side caching
    - In this case, actual pages accessed will be missing from the server side log.

Web Usage Mining VI

Pattern Discovery Activities

- Uncovering traversal patterns
  - A traversal pattern is a set of pages visited by a user in a session
  - Uncovering traversal patterns is the most common data mining technique used on clickstream data
  - Different types of traversal patterns are: Association Rules,Frequent Episodes,Sequential Patterns, Maximal Frequent Sequences, Frequent Forward Sequences.
- Features of different types of traversal patterns
  - **Duplicate** page references (backward traversals and refreshes/reloads) may or may not be allowed.
  - A pattern may be composed only of **contiguous** page references, or alternatively of any pages referenced in the same session.
  - The pattern of references may or may not be **maximal** in the session. A frequent pattern is maximal if it has no subpattern that is also frequent.

Web Usage Mining VII

TABLE 7.1: Comparison of Different Types of Traversal Patterns (from [XD01a])

	Ordering	Duplicates	Consecutive	Maximal	Support
Association rules	N	N	N	N	$\frac{\text{freq}(X)}{\#\text{ transactions}}$
Episodes	Y[1]	N	N	N	$\frac{\text{freq}(X)}{\#\text{ time windows}}$
Sequential patterns	Y	N	N	Y	$\frac{\text{freq}(X)}{\#\text{ customers}}$
Forward sequences	Y	N	Y	Y	$\frac{\text{freq}(X)}{\#\text{ forward sequences}}$
Maximal frequent sequences	Y	Y	Y	Y	$\frac{\text{freq}(X)}{\#\text{ clicks}}$

- Association Rules
  - Helps to find what pages are accessed together.
  - A page is regarded as an **item**, and a session is regarded as a **transaction**

Web Usage Mining VIII

- We can mine frequent items sets using Apriori or FP-growth algorithm.
- Example: The XYZ Corporation maintains a set of five Web pages: $\{A, B, C, D, E\}$
- Let the the following sessions (listed in timestamp order) have been created:

$$D = \{S_1 = \{U_1, (A, B, C)\}, S_2 = \{U_2, (A, C)\}, S_3 = \{U1, (B, C, E)\},$$
$$S_4 = \{U3, (A, C, D, C, E)\}\}$$

- With a support of 30%, using aptiori, the frequent itemsets are:

$$L = \{\{A\}, \{B\}, \{C\}, \{E\}, \{A, C\}, \{B, C\}, \{C, E\}\}$$

Web Usage Mining IX

Pattern Analysis

- Once patterns have been identified, they must be analyzed to determine how that information can be used.
- Some of the generated patterns may be deleted and determined not to be of **interest**.
- Examining Web logs not only to identify frequent types of traversal patterns, but also to identify patterns that are of interest because of their uniqueness or statistical properties
- Pattern can be analyzed using a Web mining query language, **MINT**
  - With MINT, selection of patterns that satisfy a $g-sequence$ template are accomplished
  - A $g-sequence$ is a vector that consists not only of the pages visited (events) but also of wildcards
  - Eg: $g-sequence\ b*c$ stands for a sequence consisting of b, any number of pages, then c.

Web Usage Mining X

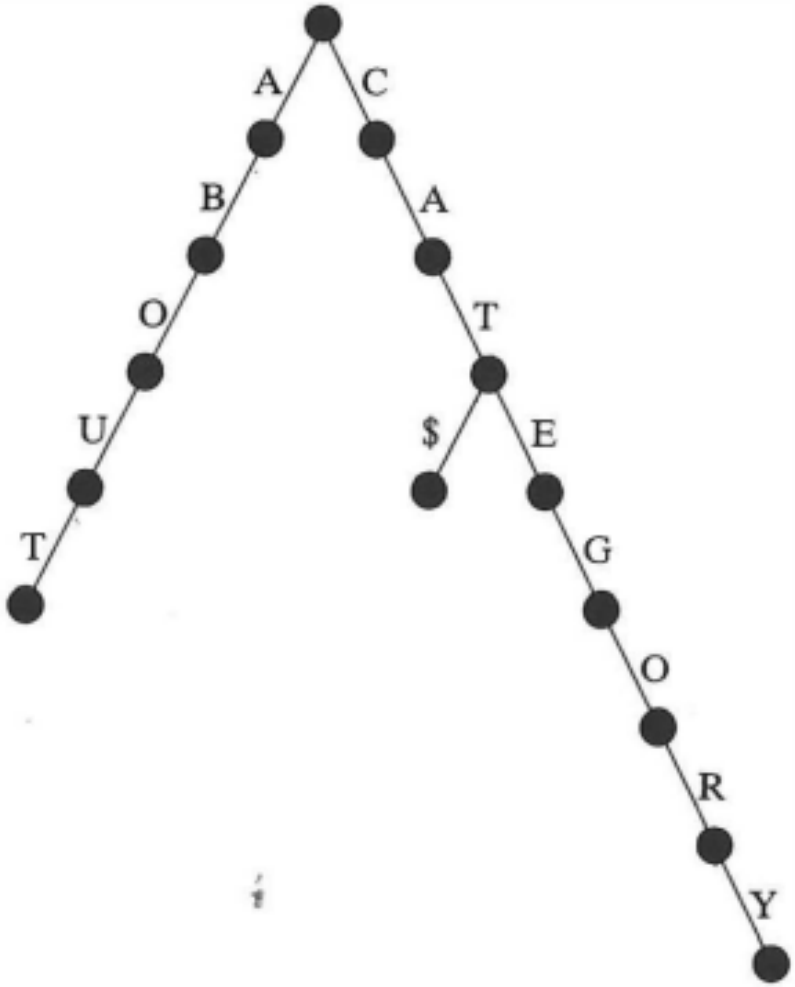
- The patterns found across the two logs are then compared for similarity
  - Two patterns are comparable if their $g - sequences$ have at least the first n pages the same. Here n is supplied by the user.
- This helps to compare the differences between traversal patterns of the customers of an e-business site and those that are not customers

Web Usage Mining XI

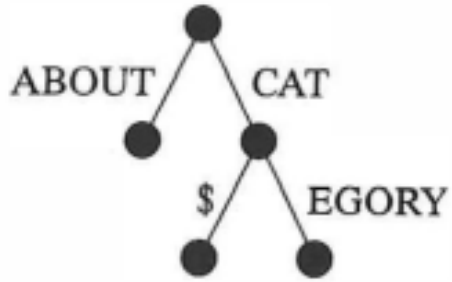
Trie: A Data Structure for tracking patterns

- A $trie$ is a rooted tree, where each path from the root to a leaf represents a sequence
- Trie's are used to store strings for pattern-matching applications.
- Each character in the string is stored on the edge to the node. Common prefixes of strings are shared.
- A problem in using tries for many long strings is the space required
- Solution: $suffix-tree$: a compressed version of the $trie$.
- Example shows a standard $trie$, $suffix-tree$ for the three strings: $\{ABOUT, CAT, CATEGORY\}$

Web Usage Mining XII



(a) Trie



(b) Suffix tree

Web Usage Mining XIII

Applications of web usage mining

- Personalization of users:
  - Keeping track of previously accessed pages of a user, identify the typical browsing behavior of a user and subsequently to predict desired pages.
- Helps to improve Performance of future access:
  - By determining frequent access behavior for users, needed links can be identified to improve the overall performance of future accesses.
- Caching: Information concerning frequently accessed pages can be used for caching.
- Helps to improve the actual design of Web pages
  - Identifying common access behaviors can be used to improve the actual design of Web pages
  - Example: e-commerce site can be identified as either customers or noncustomers by comparing behavior of customers can with that for those who do not purchase anything.

Web Usage Mining XIV

- Web usage patterns helps to gather business intelligence to improve sales and advertisement.
- Gathering statistics concerning how users actually access Web pages may or may not be viewed as part of mining

Text Mining I

- Text mining is an *interdisciplinary* field that draws on information retrieval, data mining, machine learning, statistics, and computational linguistics
- A substantial portion of *information is stored as text* such as news articles, technical papers, books, digital libraries, email messages, blogs, and web pages
- An important *goal* of text mining is to *derive high-quality information from text*
- Text mining typically done through the discovery of patterns and trends.
- Typical *text mining tasks* include: *text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summaization, and entity-relation modeling (i.e., learning relations between named entities).*

Text Mining II

Basic Measures for Text Retrieval

- Let the set of documents relevant to a query be denoted as $\{Relevant\}$
- $\{Retrieved\}$: set of documents retrieved
- $\{Relevant\} \cap \{Retrieved\}$: set of documents that are both relevant and retrieved Precision:
  - Percentage of retrieved documents that are in fact relevant to the query (i.e., "correct" responses)

$$recall = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Retrieved\}|}$$

Recall:

- Percentage of documents that are relevant to the query and were, in fact, retrieved.

$$precision = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Relevant\}|}$$

Text Mining III

F-score:

- Provides a single score that balances both the concerns of precision and recall in one number(trade-off)

$$F\_score = \frac{recall \times precision}{(recall + precision)/2}$$

Text Mining IV

Text Retrieval Methods

- Text retrieval methods fall into two categories: Document selection method, Document ranking method
- **Document Selection Methods**
  - Use queries to specify constraints for selecting relevant documents
  - Eg; Boolean retrieval model: document is represented by a set of keywords
  - And a user provides a Boolean expression of keywords, such as "car and repair shops," "tea or coffee," or "database systems $butnot$ Oracle."
  - The retrieval system would take such a Boolean query and return documents that satisfy the Boolean expression

Text Mining V

- **Document ranking methods**
  - Use the *query and rank* all documents in the order of relevance
  - It present a *ranked list* of documents in response to a user's keyword query.
  - For ranking, it *matches the keywords in a query* with those in the documents and score each document based on how well it matches the query.
  - The goal is to approximate the *degree of relevance* of a document with a score computed based on information such as the frequency of words in the document and the whole collection.
  - Different ranking methods are used including algebra, logic, probability, and statistics.
  - For ordinary users and exploratory queries, these methods are more appropriate than document selection methods.

Text Mining VI

Tokenization

- It is the first step(a preprocessing step) in most retrieval systems which identify keywords for representing documents.
- Indexing useless words is avoided using a *stop list* associated with the set of documents
- A *stop list* is a set of words that are deemed "irrelevant." Eg; *the*, *of*, *for*, *with*, and so on...

Text Mining VII

Vector Space Model

- The most popular text retrieval method
- It represent a document and a query both as *vectors* in a high-dimensional space corresponding to all the keywords
- Then use an appropriate *similarity measure to compute the similarity* between the query vector and the document vector
- The similarity values can then be used for ranking documents.
- It models a set of d documents and a set of t terms as a vector v in the t dimensional space $\mathcal{R}^t$
- Let's introduce some terminologies related to vector space model
- **term frequency, freq(d,t):**
  - the number of occurrences of term t in the document d
  - *term frequency = freq(d,t)*

Text Mining VIII

- **term-frequency matrix, TF(d,t)**:
  - measures the *association* of a term t with respect to the given document d
  - 0 if the document does not contain the term, and nonzero otherwise
  - We can set $TF(d,t) = 1$ if the term t occurs in the document d, or use the term frequency $freq(d,t)$, or the relative term frequency
  - Relative term frequency, is the term frequency versus the total number of occurrences of all the terms in the document.
  - Normalzing term frequency:

$$TF(d,t) = \begin{cases} 0 & if\, freq(d,t) = 0 \\ 1 + \log(1 + \log(freq(d,t))) & otherwise. \end{cases}$$

Text Mining IX

- **Inverse document frequency (IDF):**
  - represents the scaling factor, or the importance, of a term t.
  - If a term t occurs in many documents, its importance will be scaled down due to its reduced discriminative power.

$$IDF(t) = \log \frac{1 + |d|}{|d_t|}$$

  - d_t is the set of documents containing term t
  - If $|d_t| \ll |d|$, the term t will have a large IDF scaling factor and vice versa.
- In a complete vector-space model, TF and IDF are combined together, which forms the $TF - IDF$ measure:

$$TF - IDF(d,t) = TF(d,t) \times IDF(t)$$

Text Mining X

TF-IDF: Example 1

- Term frequency matrix given in the table shows the frequency of terms per document. Calculate the $TF - IDF$ value for the term T_6 in document d_4

A term frequency matrix showing the frequency of terms per document.

document/term	t_1	t_2	t_3	t_4	t_5	t_6	t_7
d_1	0	4	10	8	0	5	0
d_2	5	19	7	16	0	0	32
d_3	15	0	0	4	9	0	17
d_4	22	3	12	0	5	15	0
d_5	0	7	0	9	2	4	12

Text Mining XI

Solution

—- Write equations for TF, IDF here—-

$$TF(d_4, t_6) = 1 + \log(1 + \log(15)) = 1.3377$$
$$IDF(t_6) = \log \frac{1+5}{3} = 0.301$$
$$TF - IDF(d_4, t_6) = 1.3377 \times 0.301 = 0.403$$

Text Mining XII

TF-IDF: Example 2

- From the given Term frequency matrix, Calculate the $TF-IDF$ value for the term T_4 in document d_3

Document/term	T1	T2	T3	T4	T5	T6
D1	5	9	4	0	5	6
D2	0	8	5	3	10	8
D3	3	5	6	6	5	0
D4	4	6	7	8	4	4

- Solution: Write equations, calculate TF,IDF, $TF-IDF(d_3, t_4) = 0.2749$

Text Mining XIII

Text Indexing Techniques

- Popular text retrieval indexing techniques: *inverted indices, signature files*
- **Inverted indices**
  - An inverted index is an index structure that maintains two hash indexed indexed tables: *document_table* , *term_table*
  - *document_table*: a set of document records, each containing two fields: *doc_id* and *posting_list*, where *posting_list* is a list of terms (or pointers to terms) that occur in the document, sorted according to some relevance measure.
  - *term_table* consists of a set of term records, each containing two fields: *term_id* and *posting_list*, where *posting_list* specifies a list of document identifiers in which the term appears.
  - Such organization make it easy to answer queries like "*Find all of the documents associated with a given set of terms*," or "*Find all of the terms associated with a given set of documents*."

Text Mining XIV

- Here to find all of the documents associated with a set of terms, we can first find a list of document identifiers in *term_table* for each term, and then intersect them to obtain the set of relevant documents

- **Signature files**
  - Stores a signature record for each document in the database.
  - Each signature has a fixed size of b *bits representing term*
  - Each bit of a document signature is initialized to 0.
  - A bit is set to 1 if the term it represents appears in the document.
  - A signature S_1 matches another signature S_2 if each bit that is set in signature S_2 is also set in S_1
  - Since there are usually more terms than available bits, multiple terms may be mapped into the same bit.

Text Mining XV

Text Mining Approaches

- Major approaches which are based on the kinds of data they take as input: keyword-based approach, tagging approach, information-extraction approach
- **Keyword-based approach**
  - where the input is a set of keywords or terms in the documents
  - A simple keyword-based approach may only discover relationships at a relatively shallow level
  - Eg: rediscovery of compound nouns (e.g., "database" and "systems") or co-occurring patterns with less significance (e.g., "terrorist" and "explosion")
  - It may not bring much deep understanding to the text
- **Tagging approach**
  - where the input is a set of tags

Text Mining XVI

- It rely on tags obtained by *manual tagging* (which is costly and is unfeasible for large collections of documents) or by some *automated categorization algorithm*
- *automated categorization algorithm* may process a relatively small set of tags and require defining the categories beforehand

- **Information-extraction approach**
  - which inputs *semantic information*, such as events, facts, or entities uncovered by information extraction.
  - more advanced and lead to the discovery of some deep knowledge
  - it requires semantic analysis of text by natural language understanding and machine learning methods.
  - This is a challenging knowledge discovery task.
- The text mining tasks include document *clustering, classification, information extraction, association analysis, and trend analysis*

Text Mining XVII

Keyword-Based Association Analysis

- collects sets of *keywords or terms* that occur *frequently together* and then finds the *association* or correlation relationships among them.
- First *preprocess the text data* by parsing, stemming, removing stop words, and so on, and then evokes association mining algorithms
- It views each *document* as a *transaction*, while a set of *keywords* in the document can be considered as a set of *items* in the transaction.
- That is, the database is in the format: $\{document\_id, a\_set\_of\_keywords\}$
- The association mining process can help detect both *compound associations*and *noncompound associations*
- *Compound associations* include domain-dependent terms or phrases, such as [Stanford, University] or [U.S., President, George W. Bush]

Text Mining XVIII

- *Noncompound associations* such as [dollars, shares, exchange, total, commission, stake, securities].
- Mining based on these associations is referred to as "*term-level association mining*" (as opposed to mining on individual words)
- Advantages:
  - terms and phrases are automatically tagged so there is no need for human effort in tagging documents
  - the number of meaningless results is greatly reduced, as is the execution time of the mining algorithms

Text Mining XIX

Document Classification Analysis

- Document classification *organize* tremendous number of on-line *documents into classes* to facilitate document retrieval and subsequent analysis.
- Document classification has been used in many areas: automated topic tagging (i.e., assigning labels to documents), topic directory construction, identification of the document writing styles (which may help narrow down the possible authors of anonymous documents)
- Automated document classification
  - Automated document classification classifies documents automatically
  - First, a set of preclassified documents is taken as the training set.
  - The training set is then analyzed in order to derive a classification scheme
  - Refine the classification scheme with a testing process
  - The derived classification scheme can be used for classification of other on-line documents.

Text Mining XX

- The process of document classification is similar to the classification of *relational data*
- But relational data are *well structured* , while, document databases are *not structured* according to attribute-value pairs
- Therefore, commonly used relational data-oriented classification methods, such as decision tree analysis, may not be effective for the classification of document databases.
- The classification methods used in text classification are: *k-nearest-neighbor classification, Bayesian classification, support vector machines, and association-based classification*

Text Mining XXI

- **knn classification**
  - knn works based on intuition that *similar documents* are expected to be assigned the *same class label*
  - Initially, simply index all of the *training* documents, each associated with its corresponding class label
  - When a test document is submitted it is treated as a query to the IR system
    - Retrieve k documents from the training set that are most similar to the query
    - Find the*class label distribution* of the test document's k *nearest neighbors*.
    - From the class label distribution, it finds the class label of the test document
- **Bayesian classification**
  - a popular technique for effective document classification
  - First it trains the model by calculating a generative document distribution $P(d|c)$ to each class c of document d

Text Mining XXII

- Then tests which class is most likely to generate the test document
- **SVM Classification**
  - Represent classes by *numbers* and construct a *direct mapping function* from term space to the *class variable*
  - They Perform effective classification since they work well in high-dimensional space.
- **Association-based classification**
  - Classifies documents based on a set of associated, frequently occurring text patterns
  - They Perform effective classification since they work well in high-dimensional space.
  - Since *frequent terms* are *poor discriminators*, they use terms that are not very frequent(good discriminative power) for document classification
  - First, keywords and terms can be extracted by information retrieval and simple association analysis techniques

Text Mining XXIII

- Second, concept hierarchies of keywords and terms can be obtained using available term classes, such as WordNet, or relying on expert knowledge, or some keyword classification systems.
- Third, A term association mining method is applied to discover sets of associated terms that can be used to maximally distinguish one class of documents from others.
- This derives a set of association rules associated with each document class

Text Mining XXIV

Document Clustering Analysis

- Document clustering is one of the most crucial techniques for *organizing documents* in an *unsupervised manner*.
- The document space is always of very *high dimensionality*, ranging from several hundreds to thousands.
- Due to the curse of dimensionality, it makes sense to project the documents into a lower-dimensional subspace in which the semantic structure of the document space becomes *clear*
- In the low-dimensional semantic space, the traditional clustering algorithms are then be applied.
- The clustering methods include: spectral clustering, mixture model clustering, clustering using Latent Semantic Indexing, and clustering using Locality Preserving Indexing.

Text Mining XXV

- **Spectral clustering method**
  - First performs *spectral embedding* (dimensionality reduction) on the original data
  - Then applies thetraditional clustering algorithm (e.g., k-means) on the *reduced document space*.
  - It is capable of handling highly nonlinear data (the data space has high curvature at every local area)
  - Drawbacks
    - they use the nonlinear embedding (dimensionality reduction), which is only defined on "training" data
    - They have to use all of the data points to learn the embedding
    - When the data set is very large, it is computationally expensive to learn such an embedding.

Text Mining XXVI

- **Mixture model clustering method**
  - models the text data with amixture model, often involving multinomial component models
  - First, *estimate the model parameters* based on the text data and any additional prior knowledge
  - infer the clusters based on the estimated model parameters
  - One potential advantage of such clustering methods is that the clusters can be designed to *facilitate comparative analysis of documents*.

Graph Mining I

- Graphs become increasingly important in modeling complicated structures
- It can model *circuits, images, chemical compounds, protein structures, biological networks, social networks, the Web, workflows, and XML documents*.
- Graph mining is an active and important theme in data mining.
- Graph mining *discovers* various kinds of *graph patterns* such as frequent substructures
- frequent substructures are useful for
  - *characterizing* graph sets,
  - discriminating different *groups* of graphs,
  - *classifying* and *clustering* graphs,
  - building graph *indices*, and
  - facilitating similarity *search* in graph databases
- Graph mining tasks include *mining frequent subgraph patterns*, graph classification, clustering, and other analysis tasks

Graph Mining II

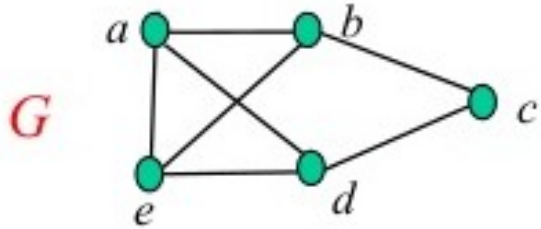
Methods for Mining Frequent Subgraphs

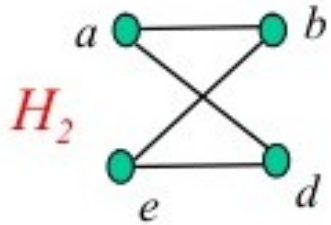
Some preliminary concepts

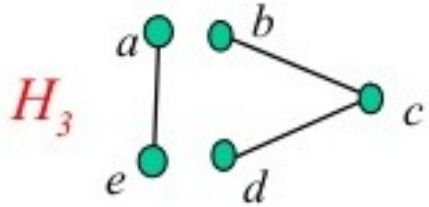
- For a given graph g, *vertex set* is denoted by $V(g)$ and the *edge set* by $E(g)$
- A label function, L, maps a vertex or an edge to a label
- A graph g' is a *subgraph* of another graph g if there exists a subgraph isomorphism from g' to g
- Formal Definition: A graph $g' = (V', E')$ is a subgraph of another graph $g = (V, E)$ iff
  - $V' \subseteq V$
  - $E' \subseteq E \wedge ((v_1, v_2) \in E' \implies v_1, v_2 \in V'$
- Given a labeled graph data set, $D = \{G_1, G_2, ..., G_n\}$, we define $support(g')$ (or $frequency(g')$) as the percentage (or number) of graphs in D where g' is a subgraph

Graph Mining III

- Example: H_1, H_2, and H_3 are subgraphs of G

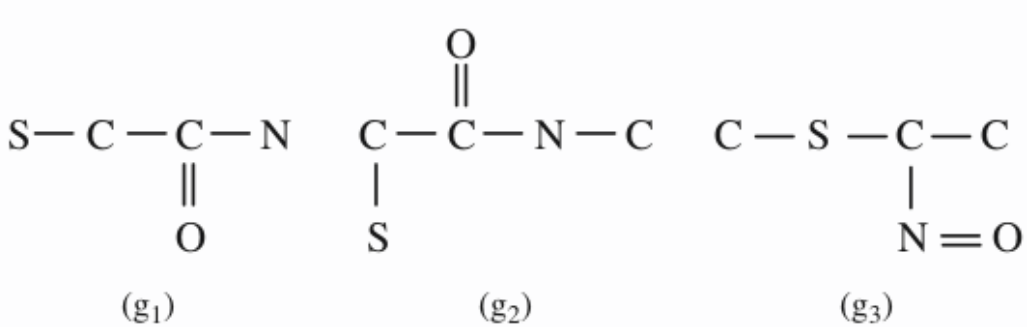




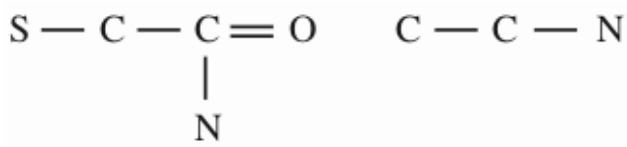


Graph Mining IV

- Frequent graph: a graph whose support is *no less than* a minimum support threshold, min_sup.
- sample set of chemical structures, and two of the frequent subgraphs with $min\_sup = 66.6\%$.



A sample graph data set.



Frequent graphs.

Graph Mining V

Apriori-based Approach for Mining Frequent Subgraphs

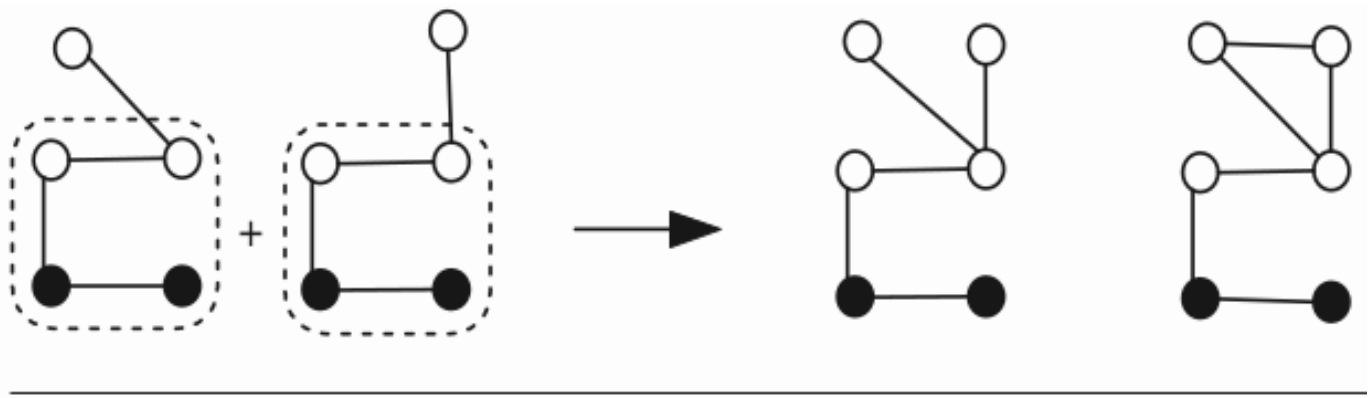
- It share *similar characteristics* with Apriori-based frequent itemset mining algorithms
- The searching starts with graphs of small size (like candidate 1-itemsets(C_1) in itemset mining)
- The finds the frequent substructures(S_1) that satisfy min_sup (like L_1 in itemset mining)
- These new candidate substructures($(C_2, C_3...)$) are generated by joining two similar but slightly different frequent subgraphs that were discovered in the previous step.
- Then proceeds in a bottom-up manner by generating candidates $(C_2, C_3...)$ having an extra vertex, edge, or path
- At each step, it discovers new frequent substructure(S_k) of size k which satisfy the min_sup (like $L_1, L_2, ...$ in itemset mining)

Graph Mining VI

- The candidate generation problem in frequent substructure mining is harder than that in frequent itemset mining, because there are many ways to join two substructures.
- The methods used for candidate generation are: AGM, FSG, and a path-join method.
- AGM algorithm
  - AGM shares similar characteristics with Apriori-based itemset mining
  - It uses a vertex-based candidate generation method that increases the substructure size by one vertex at each iteration of *AprioriGraph* algorithm
  - Two $size - k$ frequent graphs are joined only if they have the same $size - (k-1)$ subgraph
  - Here, *graph size* is the number of vertices in the graph
  - Figure below depicts the two substructures joined by two chains (where a chain is a sequence of connected edges).
  - The newly formed candidate includes:

Graph Mining VII

- the $size-(k-1)$ subgraph in common(dashed boxes) and the additional two vertices from the two $size-k$ patterns (vertices outside dashed box)
-
- has one more vertex than these two patterns

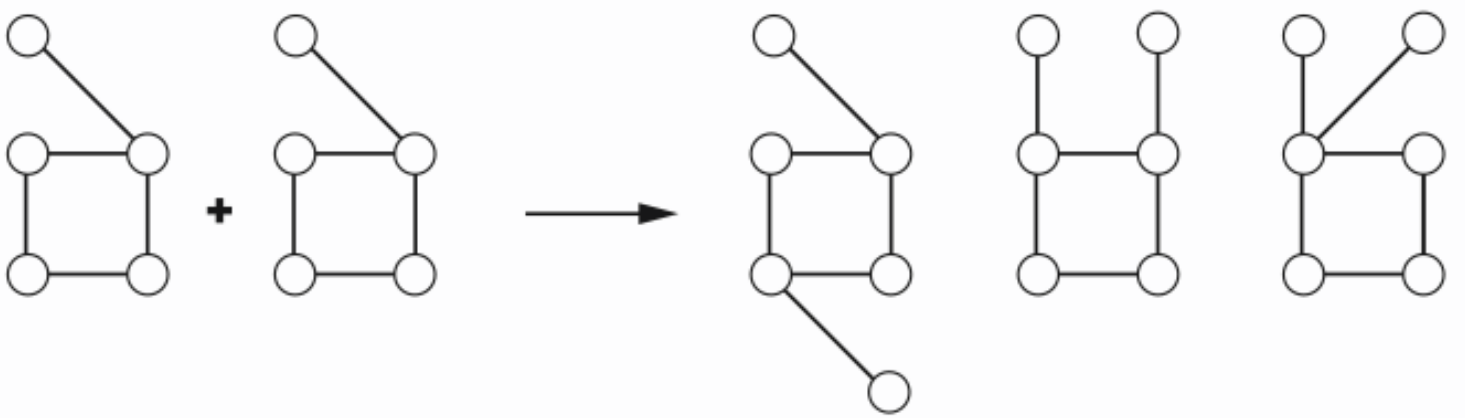


AGM: Two substructures joined by two chains.

- FSG algorithm
  - FSG method adopts an edge-based candidate generation
  - ie; it explore edges and connections in an Apriori-based fashion
  - This strategy increases the substructure size by one edge in each call of *AprioriGraph*

Graph Mining VIII

- Two $size - k$ patterns are merged if and only if they share the same subgraph having $k - 1$ edges, which is called the $core$
- Here, *graph size* is taken to be the number of edges in the graph.
- Figure shows potential candidates formed by two structure patterns
- The newly formed candidate include:
  - the $core$ and the additional two edges from the $size - k$ patterns.
  - has one more edge than these two patterns



FSG: Two substructure patterns and their potential candidates.

- Path-join method

Graph Mining IX

- It also explore edges and connections in an Apriori-based fashion
- It is an edge-disjoint path method
- Here the graphs are classified by the number of disjoint paths they have
- Two paths are edge-disjoint if they do not share any common edge
- A substructure pattern with $k+1$ disjoint paths is generated by joining substructures with k disjoint paths.

Graph Mining X

the Algorithm

Method:

$S_1 \leftarrow$ frequent single-elements in the data set;
Call AprioriGraph(D, min_sup, S_1);

procedure AprioriGraph(D, min_sup, S_k)

(1) $S_{k+1} \leftarrow \varnothing$;
(2) **for each** frequent $g_i \in S_k$ **do**
(3) **for each** frequent $g_j \in S_k$ **do**
(4) **for each** size $(k+1)$ graph g formed by the merge of g_i and g_j **do**
(5) **if** g is frequent in D and $g \notin S_{k+1}$ **then**
(6) insert g into S_{k+1};
(7) **if** $s_{k+1} \neq \varnothing$ **then**
(8) *AprioriGraph*(D, min_sup, S_{k+1});
(9) **return**;

Graph Mining XI

Pattern-Growth Approach for Mining Frequent Subgraphs

- The pattern-growth approach is more flexible regarding its search method
- It can use breadth-first search(BFS) as well as depth-first search (DFS) strategy while Apriori use BFS only
- It works based on extending the graph g by adding a new edge e
  - The newly formed graph is denoted by $g \diamond_x e$
  - Edge e may or may not introduce a new vertex to g
  - If e introduces a new vertex, we denote the new graph by $g \diamond_{xf} e$, otherwise,$g \diamond_{xb} e$
  - where f or b indicates that the extension is in a $forward$ or $backward$ direction.
- We refer to the algorithm as *PatternGrowthGraph*
- For each discovered graph g, it performs extensions recursively until all the frequent graphs with g embedded are discovered.

Graph Mining XII

- The recursion stops once no frequent graph can be generated.
- PatternGrowthGraph is simple, but not efficient. Since, the same graph can be discovered many times due to recursive extensions
- We call a graph that is discovered a second time a *duplicate graph*.
- *gSpan algorithm*: An algorithm designed to reduce the generation of duplicate graphs.

Graph Mining XIII

the Algorithm

Output:

- The frequent graph set, S.

Method:

$S \leftarrow \varnothing$;
Call PatternGrowthGraph(g, D, min_sup, S);

procedure PatternGrowthGraph(g, D, min_sup, S)

(1) **if** $g \in S$ **then return;**
(2) **else** insert g into S;
(3) scan D once, find all the edges e such that g can be extended to $g \diamond_x e$;
(4) **for each** frequent $g \diamond_x e$ **do**
(5) *PatternGrowthGraph*($g \diamond_x e$, D, min_sup, S);
(6) **return;**

Social Network Analysis-Concepts I

- It is the process of investigating social structures through the use of networks and graph theory
- It characterizes networked structures in terms of nodes (individual actors, people, or things within the network) and the ties, edges, or links (relationships or interactions) that connect them
- From a data mining perspective, is also called link analysis or link mining

Social Network Analysis-Concepts II

- From the data mining point of view, a social network is a *heterogeneous* and *multirelational data* set represented by a graph
- The graph is typically very large, with nodes corresponding to objects and edges corresponding to links representing relationships or interactions between objects
- Both nodes and links have attributes. Objects may have class labels.
- Links can be one-directional and are not required to be binary.
- Social networks are rarely static. Their graph representations evolve as nodes and edges are added or deleted over time.
- Social networks need not be social in context.
- Eg; *Electrical power grids, telephone call graphs, the spread of computer viruses, the World Wide Web, and coauthorship and citation networks of scientists.*

Social Network Analysis-Concepts III

Characteristics of Social Networks

Measures used to characterize Social Networks

- Nodes degree: the number of edges incident to each node
- Distance between nodes: shortest path length between a pair of nodes
- Network diameter: the maximum distance between pairs of nodes
- Average distance: Average of pairwise node-to-node distances
- Effective diameter: the minimum distance, d, such that for at least 90% of the reachable node pairs, the path length is at most d

Characteristics

1. **Densification power law or growth power law:**
   - "*Networks become increasingly dense over time with the average degree increasing*"
   - Hence, the *number of edges growing superlinearly* in the number of nodes
   - Densification power law can be stated as: $e(t) \propto n(t)^a$

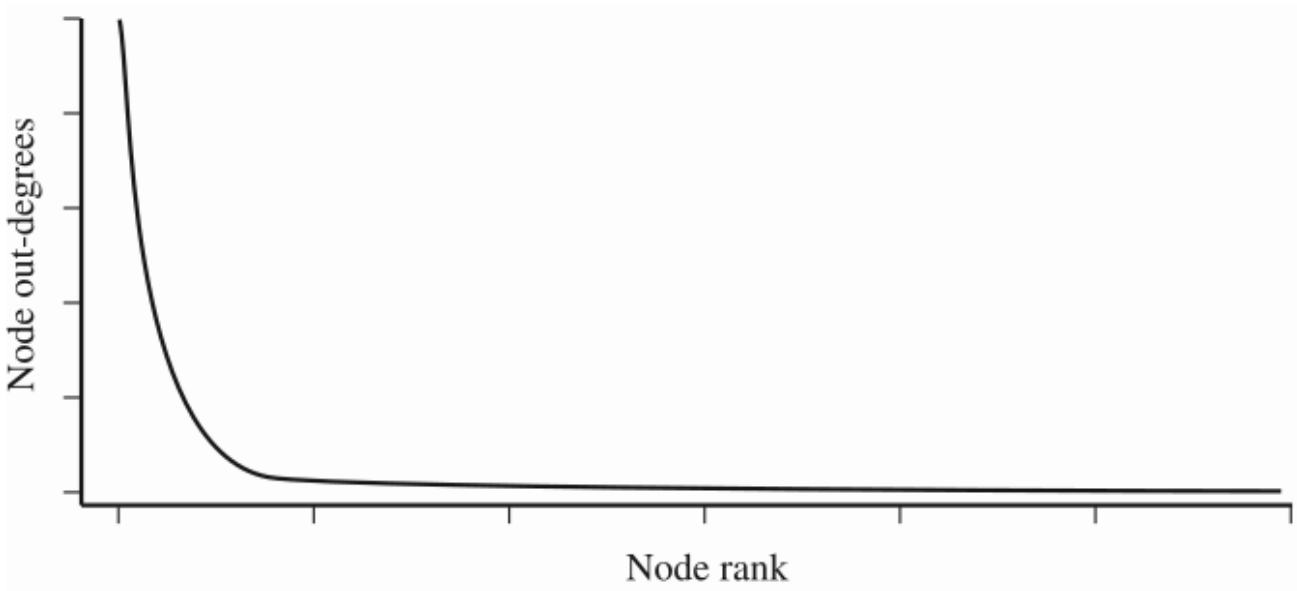
Social Network Analysis-Concepts IV

- $e(t)$: number of edges of graph at time t
- $n(t)$: number of nodes of graph at time t
- a : exponent generally lies strictly between 1 and 2. If $a = 1$,corresponds to constant average degree over time, If $a = 2$ corresponds to an extremely dense graph

- **Shrinking diameter:**
  - The *"effective diameter tends to decrease as the network grows".*
  - Example: citation network
    - Where nodes are papers and a citation from one paper to another is indicated by a directed edge.
    - The out-links of a node, v (representing the papers cited by v), are "frozen" at the moment it joins the graph
    - The decreasing distances between pairs of nodes consequently appears to be the result of subsequent papers acting as "bridges" by citing earlier papers from other areas.

Social Network Analysis-Concepts V

- **Heavy-tailed out-degree and in-degree distributions:**
  - The number of out-degrees and in-degrees for a node tends to follow a heavy-tailed distribution by observing the power law, $1/n^a$
  - where n is the rank of the node in the order of decreasing out-degrees and typically, $0 < a < 2$
  - The smaller the value of a, the heavier the tail.



Link Mining(Social Network Mining) I

Tasks

1. Link-based object classification

- In traditional classification methods, objects are classified based on the attributes that describe them
- Link-based classification predicts the category of an object based not only on its attributes, but also on its links, and on the attributes of linked objects.
- Example 1: *Web page classification*
  - A well-recognized example of link-based classification
  - It predicts the category of a Web page based on *word occurrence* and *anchor text* (attributes) and *links* between pages and *other attributes* of the pages and links
  - *Word occurrence* is the words that occur on the page,
  - *Anchor text* is the hyperlink words, that is, the words you click on when you click on a link
- Example 2: Bibliography domain objects-papers,authors,journals,etc.

Link Mining(Social Network Mining) II

- predict the topic of a paper based on *word occurrence, citations* (other papers that cite the paper), and cocitations (other papers that are cited within the paper), where the citations act as links.

- Example 3: Eidemiology domain object - patient
  - Predicts the disease type of a patient based on characteristics (e.g., *symptoms*) of the patient, and on *characteristics* of other people with whom the patient has been in *contact*

2. Object type prediction
   - Predicts the type of an object, based on its attributes and its links, and on the attributes of objects linked to it
   - Example 1: Bibliography domain- predict the venue type of a publication as either conference, journal, or workshop
   - Example 2: Communication domain- predict whether a communication contact is by e-mail, phone call, or mail.

Link Mining(Social Network Mining) III

3. Link type prediction
   - predicts the type or purpose of a link, based on properties of the objects involved.
   - Example 1: Epidemiological domain
     - predict whether two people who know each other are family members, coworkers, or acquaintances
     - predict whether there is an advisor-advisee relationship between two coauthors.
   - Example 2: Web domain - pedict whether a link on a page is an advertising link or a navigational link.
4. Predicting link existence
   - predict whether a link exists between two objects
   - Example 1: Web domain- predicting whether there will be a link between two Web pages
   - Example 2: Bibliography domain- predicting whether a paper will cite another paper

Link Mining(Social Network Mining) IV

- Example 3: Epidemiology domain- predict with whom a patient came in contact

5. Link cardinality estimation
   - Estimates two form of link cardinality
   1. In-links: predict the number of links to an object
      - Useful in predicting the authoritativeness of a Web page based on the number of links to it
   2. Out-links: predict the number of out links from an object
      - Useful identify Web pages that act as *hubs*, where a *hub* is one or a set of Web pages that point to many authoritative pages of the same topic
   - Example 1: Bib domain- the number of citations in a paper may indicate the impact of the paper—the more citations the paper has, the more influential it is likely to be
   - Example 2: Epidemiology domain- predict the number of links between a patient and his or her contacts is an indication of the potential for disease transmission.

Link Mining(Social Network Mining) V

6. Object reconciliation

- predict whether two objects are, in fact, the same, based on their attributes and links.
- This task is common in information extraction, duplication elimination, object consolidation, and citation matching, and is also known as record linkage or identity uncertainty.
- Example 1: Web domain- predicting whether two websites are mirrors of each other
- Example 2: Bib domain- predicting whether two citations actually refer to the same paper
- Example 3: Epidemiology domain predicting whether two apparent disease strains are really the same.

7. Group detection

- predicts when a set of objects belong to the same group or cluster, based on their attributes as well as their link structure.
- Example 1: Web domain- identification of Web communities, which is a collection of Web pages that focus on a particular theme or topic

Link Mining(Social Network Mining) VI

- Example 2: Bib domain identification of research communities

8. Subgraph detection
   - finds characteristic subgraphs within networks
   - Example 1: Biology domain- discovers subgraphs corresponding to protein structures
   - Example 2: Chemistry domain - search for subgraphs representing chemical substructures.
9. Metadata mining
   - Mines metadata information
   - Metadata provide semi-structured data about unstructured data, ranging from text and Web data to multimedia databases.
   - Example 1: Web or Database Domain
     - *schema mapping* - identify whether attribute $customer\_id$ from one database is mapped to $cust\_number$ from another database
     - *schema discovery*- which generates schema from semi-structured data
     - *schema reformulation* - which refines the schema based on the mined metadata.

Link Mining(Social Network Mining) VII

- Example 2: Bib domain- helps in matching two bibliographic sources
- Example 3: Epidemiology domain- helps to map between two medical ontologies.

Link Mining(Social Network Mining) VIII

Challenges

1. Logical versus statistical dependencies
   - Two types of dependencies reside in the graph: *link structures, probabilistic dependencies*
   - *link structures*: represents the logical relationship between objects
   - *probabilistic dependencies*: represents statistical relationships, such as correlation between attributes of objects where, typically, such objects are logically related
   - Coherent handling of these dependencies is a challenge for multirelational data mining, where the data to be mined exist in multiple tables.
   - We must search over the different possible logical relationships between objects, in addition to the standard search over probabilistic dependencies between attributes.
   - This takes a huge search space, which further complicates finding a plausible mathematical model.

Link Mining(Social Network Mining) IX

2. Feature construction
   - In link-based classification, we consider the attributes of an object as well as the attributes of objects linked to it
   - The goal of *feature construction* is to construct a single feature representing these attributes
   - This construction involve *feature selection* and *feature aggregation*.
   - *Feature selection*, selects only the most discriminating features.
   - *Feature aggregation* takes a multiset of values over the set of related objects and returns a summary of it.
   - This summary may be: the *mode* (most frequently occurring value); the *mean value* of the set (if the values are numerical); or the *median* or "*middle*" value (if the values are ordered).
   - However, in practice, this method is not always appropriate.
3. Instances versus classes
   - It alludes (can not identify) whether the model refers explicitly to *individuals*(instance) or to *classes* (generic categories) of individuals.

Link Mining(Social Network Mining) X

4. Collective classification and collective consolidation
   - Traditional classifier models are trained based on set of class-labeled objects and consider only the attributes of the objects.
   - When new set of unlabeled objects comes in, the prediction will be complicated doe to possible correlations between objects
   - ie; the labels of linked objects may be correlated
   - To avoid this, a consolidation should take place to update the class label of each object based on the labels of objects linked to it.
   - In this sense, classification is done *collectively* rather than independently.
5. Effective use of labeled and unlabeled data
   - Links between labeled (training) data and unlabeled (test) data induce dependencies that can help make more accurate inferences.
6. Closed versus open world assumption
   - Most traditional approaches assume that we know all the potential entities in the domain.

Link Mining(Social Network Mining) XI

- This "closed world" assumption is unrealistic in real-world applications

7. Community mining from multirelational networks
   - community mining: discovers groups of objects that share similar properties
   - *Web page linkage* is an example, where a discovered community may be a set of Web pages on a particular topic.
   - Most algorithms for community mining assume that there is only one social network, representing a relatively homogenous relationship.
   - In reality, there exist multiple, heterogeneous social networks, representing various relationships(multirelational)
   - A new challenge is the mining of hidden communities on such heterogeneous social networks


thank
you!